User Manual for Software Package **CAP** – a **C**ontinuous **A**rray **P**rocessing Toolkit for Ambient Vibration Array Analysis

written by Matthias Ohrnberger[1]

Contributions by (in alphabetical order)

Sylvette Bonnefoy-Claudet[2], Cecile Cornou[3], Bertrand Guillier[4], Fortunat Kind[3], Andreas Koehler[1]
Estelle Schissele-Rebel[1], Alexandros Savvaidis[5], Marc Wathelet[6]

[1]Institute of Geosciences, University of Potsdam, Germany
[2]LGIT, Universite Joseph Fourier, Grenoble, France
[3]Swiss Seismological Survey, ETH Zuerich, Switzerland
[4]IRD, Grenoble, France
[5]Institute of Engineering Seismology and Earthquake Engineering (ITSAK), Thessaloniki, Greece
[6]GEOMAC, Universite de Liege, Liege, Belgium

July 12, 2004

Contents

1 Introduction **4**
1.1 Purpose 4
1.2 Concept 5

2 Basic principles of array techniques **7**
2.1 f-k spectrum 7
2.2 CVFK 9
2.3 CAPON 9
2.4 MUSIC 10
2.5 MSPAC method 12

3 Database connectivity **16**
3.1 **CAP** and GIANT 16
3.2 **CAP** and GEOPSY (former SARDINE) 17
3.3 **CAP** and FAKE_DB 17

4 Preprocessing Block **18**
4.1 Integer Decimation 18
4.2 Butterworth Bandpass Filtering 18
4.3 Instrument simulation 19
4.4 Additional processing settings 19

5 Main Processing Block **21**
5.1 Determination of time-frequency tiling 21
5.2 Determination of f-k grid layout 25
5.3 Conventional frequency wavenumber analysis – CVFK 27
5.3.1 Semblance based estimates for individual time windows - CVFK 27
5.3.2 Averaged cross spectral matrix approach - CVFK2 28
5.3.3 Gridless semblance maximization - CVFK_FAST 29
5.4 Slantstack Analysis – SLANTSTACK 30
5.5 Capon's high-resolution frequency wavenumber analysis – CAPON 30
5.6 MUltiple SIgnal Classification – MUSIC 31
5.7 Modified SPatial AutoCorrelation – MSPAC 32

5.8 Single station H/V ratio computation . . . 33
5.9 Supplemental and Experimental Methods . . . 34
5.9.1 Hypothesis testing for pre-selection – HYPTEST . . . 34
5.9.2 Cross-Correlation Stack – CCSTACK . . . 36
5.9.3 Attenuation estimation – QEST . . . 37

6 Postprocessing **39**
6.1 Slowness response evaluation (SLOWRESP) . . . 39
6.2 Determination of dispersion curves - fk2disp . . . 40
6.3 Using MAPFRAC for uncertainty bounds . . . 42

7 Usage **45**
7.1 Input files . . . 45
7.1.1 Supported waveform file formats . . . 45
7.1.2 Waveform list and station file (FAKE_DB only) . . . 45
7.1.3 Configuration file . . . 46
7.2 Output files . . . 52
7.2.1 The .tfbox file - keeping track of analysed data . . . 53
7.2.2 The .max file - main output file . . . 54
7.2.3 The .stmap file - slowness maps . . . 58
7.2.4 The .best file - enable statistics . . . 60
7.2.5 The .csh file - plotting your results . . . 60
7.2.6 Outputs on stderr and stdout . . . 61
7.3 Examples . . . 61
7.3.1 Command line usage with GIANT . . . 61
7.3.2 Command line usage with FAKE_DB . . . 67
7.3.3 GUI-interface with GEOPSY DB . . . 68
7.3.4 Command line interface with GEOPSY DB . . . 71

8 Future developments **73**

9 About ... **74**
9.1 Copyright . . . 74
9.2 Funding . . . 74
9.3 Acknowledgments . . . 74

10 References **75**

A Sample configuration file **77**

List of Figures

1 Example of time-frequency tiling - I 23
2 Example of time-frequency tiling - II 23
3 Example of time-frequency tiling - III 24
4 Example of time-frequency tiling - IV 24
5 Example of time-frequency tiling - V 25
6 Sampling the wavenumber grid 26
7 Cross correlation stacks for Pulheim - I 38
8 Cross correlation stacks for Pulheim - II 38
9 Example for MAPFRAC 43
10 CVFK result 63
11 CVFK_FAST result 64
12 CVFK2 result 65
13 CAPON result 65
14 MUSIC2 result 66
15 MUSIC result 66
16 MSPAC result 67
17 Startup screen of **CAP** - GEOPSY version 69
18 Selecting existing groups for processing 69
19 Specifying start and end times 70
20 Selecting a configuration file 70
21 Selecting directory for output files 71

1 Introduction

The damage produced during moderate and large earthquakes is significantly influenced by the shallow subsurface geology and soil conditions. Thus, the degree of shake-ability of the ground during strong ground motion at locations with high vulnerability has been a matter of growing interest in seismological investigations dealing with seismic hazard analysis. Site amplifications in shallow unconsolidated sediments can be predicted using the theory of linear elastic wave propagation and computing S-wave resonances due to reverberation of seismic energy between the free surface and the sediment structure overlying bedrock. The knowledge of the shallow shear wave velocity structure is essential for reasonable strong motion predictions at a given site.

Active seismic experiments and geotechnical information obtained from boreholes provide high-quality information about the shallow subsurface structure. The cost of these methods, however, is high and within densely populated urban environments, usually regions of high vulnerability, sometimes not even feasible. Since early work in the 1950's by Japanese seismologists (e.g. Aki, 1957), the use of passive, non-destructive, seismological investigation of the shallow subsurface structure has been considered as a low-cost alternative to active seismic investigation methods especially in urban environments.

Besides the widely used single-station analysis method, known as H/V ratio or Nakamura's method (for a review, see Bard, 1998), the use of small-aperture arrays allows to derive frequency dependent estimates of the phase velocity of the seismic wavefield. At most places we observe dispersion of this phase velocity curves, a property which is attributed to the surface wave part of the seismic wavefield. The dispersion curve information can be used to derive velocity models for a given site in a inversion procedure.

The extraction of dispersion curve information from ambient vibration array recordings and the subsequent inversion for the shallow shear wave velocity structure especially for sites within urban areas has been the subject of Task B (WP05-07) of the European Community financed project SESAME (Site EffectS ASsessment using AMbient Excitation, EU-Grant No.: EVG1-2000-00026). The software package **CAP** has been developed within the scope of this project in order to respond to the need of testing the potential of various frequency wavenumber techniques as well as the applicability of the spatial autocorrelation method for the extraction of phase velocity curves from microtremor array recordings.

1.1 Purpose

The software package **CAP** has been developed to allow a comparative study of the potential of different array analysis methods (both frequency wavenumber and stochastic methods) within the context of ambient vibration measurements. Although the implemented algorithms are well established quasi-standard analysis tools for seismological investigations, their use for the application domain in our focus (phase velocity curve determination from ambient vibration array recordings) is still debated.

Due to the goal of SESAME project, the purpose of **CAP** lies in the analysis of the surface wave part of the ambient vibration wavefield and the extraction of dispersion curves from the analysis results. Nevertheless, the algorithms are implemented such, that it is straightforward to use this software package for general continuous computation of wave propagation parameters in the context of seismological array analysis.

1.2 Concept

In order to allow consistent processing of microtremor array data sets we have tried to make the processing as transparent as possible. However, we did not restrict the choice of processing options or the amount of flexibility which we felt might by needed for special data sets or applications other than ambient noise analysis. Additionally, **CAP** has grown over time. As a result of this continuing proc(gr)ess, in its current stage, **CAP** is not as user-friendly as it could be ... We hope that with a more wide-spread usage of this software package and the reception of constructive feedback comments, the handling will improve.

In its current version **CAP** relies on the existence of a waveform database which allows to manage continuously recorded large data sets. Three different versions of **CAP** exist at the moment. All versions contain the same processing capabilities but differ in the I/O concept related to the underlying database structures. The different versions can be obtained from compiling the program code with different define switches and linking against different libraries. Further information is provided in section 3.

The program flow in **CAP** is divided into several blocks. After program start, user selectable parameters are read from a simple ascii file (see section 7.1.3 of this manual). A cross check is performed on the given parameter settings in order to avoid unreasonable combinations of parameters or the misuse of certain methods. If the cross-check phase has passed, the waveform data is extracted from the database followed by another cross-check of data consistency (data gap detection, changes of sampling rates, availability of data window, etc.). *Please note, that the cross-checks are not 100% safe - errors may still occurr due to unexpected combinations of parameters or inconsistent data sets.*

After these initialization steps, the preprocessing block is entered. Dependent on the user's settings, **CAP** allows for a limited number of preprocessing options applied to the raw waveform data (compare section 4). Once the preprocessing is finished, the processing loop is entered (section 5).

The processing loop is initialized by allocating memory for common data structures and pre-computation of time independent parameters derived from the settings given in the configuration file. Especially the tiling of "time-frequency boxes" as well as the sampling in the wavenumber domain (for f-k methods) are pre-built at this stage (sections 5.1 and 5.2). Depending on the selected method (sections 5.3 to 5.7), either a sliding window processing is performed (CVFK, MUSIC or MSPAC) or a averaging approach assuming signal stationarity is taken (CAPON, CVFK2, MUSIC2, SLANTSTACK[1] and HTOV).

[1] no longer implemented in the current version

After all available data has been processed, the raw analysis results are written to output files (section 7.2). In order to allow a quick visualization, a shell script is additionally created which scans the output files and creates postscript figures using the GMT software package by Wessel and Smith (1998).

2 Basic principles of array techniques

2.1 The frequency-wavenumber power spectrum - f-k spectrum

Let us consider an array of N sensors at positions $\vec{r}_n$, $(n = 1, \ldots, N)$ recording a set of q, $q < N$ uncorrelated plane waves $s_j(t), j = 1, \ldots, q$ propagating in a homogeneous medium. The time signal $x(t)$ recorded at station n located at position $\vec{r}_n$ is obtained through the superposition of the individual plane waves as:

$$x(\vec{r}_n, t) = \sum_{j=1}^{q} s_j(t + \tau_j) + \eta(\vec{r}_n, t) \tag{1}$$

where τ_j is the time delay to each of the sensors with respect to the time arrival of the wave at a reference sensor (or center of gravity of the sensor array) and $\eta_j(\vec{r}_n, t)$ stands for the uncorrelated "non-signal" part of the wavefield[1]. In frequency domain, equation (1) becomes:

$$X(\vec{r}_n, \omega) = \sum_{j=1}^{q} S_j(\omega) e^{i(\omega\tau_j)} + \eta(\vec{r}_n, \omega) \tag{2}$$

where $\omega = 2\pi f$ is the circular frequency. For a plane wave we have $\omega\tau_j = \vec{k}_j\vec{r}_n$ and $\vec{k}_j$ represents the wave number of the plane wave s_j.

The array output is defined as a multi-channel delay and sum filter operation, written in time domain as

$$y(t) = \sum_{n=1}^{N} w_n(t) x(\vec{r}_n, t - \tau_n) \tag{3}$$

where $w_n(t)$ are some weighting factors and τ_n time are delays with a reference as introduced above. In the frequency domain, the array response function becomes

$$Y(\omega) = \sum_{n=1}^{N} W_n(\omega) X(\vec{r}_n, \omega) e^{-i(\omega\tau_n)} \tag{4}$$

where $\omega\tau_n = \vec{k}\vec{r}_n$.

Using equations (2) and (4), and writing the delay times as function of wavenumber $\vec{k}$, the output of the array in the frequency-wavenumber domain is thus given by:

$$Y(\vec{k}, \omega) = \sum_{n=1}^{N} \sum_{j=1}^{q} W_n(\omega) S_j(\omega) e^{i(\vec{k}_j - \vec{k})\vec{r}_n} + \sum_{n=1}^{N} \eta(\vec{r}_n, \omega) \tag{5}$$

[1] We don't to use the common term "noise" at this point to avoid confusion with the application domain which is still sometimes called ambient seismic noise. An excellent short note about the usage of the term "noise" has been given by Scales and Snieder (1998).

An estimate of the wave propagation parameters $(\vec{k},\omega)$ is thus obtained by maximizing the modulus of $Y(\vec{k},\omega)$ within the frequency-wavenumber plane, that is $\vec{k}_j - \vec{k} = 0$.

The cross-spectrum of the recorded signals in the frequency-wavenumber domain, usually called the f-k cross-spectrum, is defined by:

$$P(\vec{k},\omega) = \sum_{l=1}^{N}\sum_{n=1}^{N}\sum_{j=1}^{q} W_n(\omega)W_l(\omega)S_{jn}(\omega)S_{jl}^{\star}(\omega)e^{i(\vec{k}_j-\vec{k})(\vec{r}_n-\vec{r}_l)} + \sum_{l=1}^{N}\sum_{n=1}^{N}\eta(\vec{r}_n,\omega)\eta^{\star}(\vec{r}_l,\omega) \quad (6)$$

where $S_{jn}(\omega), S_{jl}(\omega)$ denote the Fourier spectra of the wave s_j at receivers $\vec{r}_n$ and $\vec{r}_l$ and * symbolizes the conjugate complex.

Letting $S_{jn}(\omega) = S_{jl}(\omega) = 1$ and neglecting the uncorrelated noise, one can define the normalized beampattern $B(\vec{k},\omega)$ of the array configuration for a single plane wave incident from below by setting $\vec{k}_j = 0$ in equation 6:

$$B(\vec{k},\omega) = \frac{1}{N^2}\sum_{l=1}^{N}\sum_{n=1}^{N} W_n(\omega)W_l(\omega)e^{i\vec{k}(\vec{r}_n-\vec{r}_l)} \quad (7)$$

In matrix notation, equation (5) may be rewritten as follows

$$\underline{Y} = \underline{AWX} \quad (8)$$

where

$$\begin{aligned}
\underline{W} &= \begin{bmatrix} W_1(\omega) & 0 & \ddots & \ddots & 0 \\ 0 & W_2(\omega) & 0 & \ddots & \ddots \\ \ddots & 0 & \ddots & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & W_N(\omega) \end{bmatrix} \\
\underline{A} &= \left[e^{-i\vec{k}\vec{r}_1}, e^{-i\vec{k}\vec{r}_2}, \ldots, e^{-i\vec{k}\vec{r}_N}\right] \\
\underline{X} &= [X_1(\omega), X_2(\omega), \ldots, X_N(\omega)]^T
\end{aligned}$$

The frequency-wavenumber (f-k) cross-spectrum expressed in Matrix notation is then

$$\underline{P} = \underline{AWRW}^H\underline{A}^H \quad (9)$$

where $\underline{R} = E[\underline{XX}^H]$ is the $N \times N$ cross spectral matrix (CSM) and ^{H} denotes the hermitian conjugate operator. The cross spectral matrix is evaluated using frequency or spatial smoothing.

Equation 9 is the root of any f-k based array technique: the CSM matrix carries indeed all the informations about the propagation parameters of the waves propagating across the array; $\underline{W}$ is composed of the filter weights that can be designed in order to minimize the energy leakage from regions outside the actual signal wavenumber and $\underline{A}$ is the steering vector that is used for sweeping the wavenumber domain.

In practice one seeks the maxima of equation 9 by performing a grid-search in the wavenumber domain for a frequency f of interest. From the wavenumbers $\vec{k}_n = (k_x, k_y)_n$ of local maxima in the wavenumber map, the directions θ_n and apparent velocities $c_n(\omega)$ can be determined by:

$$\theta_n = \arctan(\frac{k_{xn}}{k_{y_n}}) \quad \text{and} \quad c_n(\omega) = \frac{\omega}{|\vec{k}_n|}$$

2.2 Conventional f-k analysis (CVFK)

For the conventional f-k analysis (CVFK), the weighting functions are set to $W_n(\omega) = 1$ and thus the f-k density cross-spectrum reduces to

$$P(\vec{k},\omega) = \frac{1}{N^2}\sum_{l=1}^{N}\sum_{n=1}^{N}\sum_{j=1}^{q} S_{jn}(\omega)S_{jl}^{\star}(\omega)e^{i(\vec{k}_j-\vec{k})(\vec{r}_n-\vec{r}_l)} + \sum_{l=1}^{N}\sum_{n=1}^{N}\eta_l^H(\vec{r}_i,\omega)\eta_n(\vec{r}_i,\omega) \tag{10}$$

The conventional estimator is then written in matrix notation

$$\underline{P}_{CV} = \underline{ARA}^H \tag{11}$$

Since the weightings are constants, the performance of the CVFK analysis is completely governed by the shape of the array beampattern at a given frequency, i.e. mainly by the array geometry. The array performance is restricted to the following wavenumber range $|\vec{k}| \in [2\pi/d_{max}, \pi/d_{min}]$, where d_{max} is the aperture of the array and d_{min} is the minimum distance between two neighbouring sensors. $2\pi/d_{max}$ is the Rayleigh limit of the array that defines the capability of the array to resolve two waves propagating at close wavenumbers and π/d_{min} is the Nyquist wavenumber.

2.3 Capon's analysis (CAPON)

Capon et al. (1967) and Capon (1969) modified the weighting functions $W_n(\omega)$ in order to minimize the f-k cross-spectrum energy carried by wavenumbers differing from the true signal wavenumber. Expressed in other words, the $W_n(\omega)$ are optimized by minimizing the signal power $\underline{WRW}^H$ except at the actual wavenumber. This last constraint is such as the array output at a given receiver is identical to the signal actually recorded at this sensor location:

$$Y_n(\omega) = W_n(\omega)X(\vec{r}_n,\omega)e^{-i\vec{k}\vec{r}_n} = X(\vec{r}_n,\omega) \tag{12}$$

resulting into the constraint

$$W_n(\omega)e^{-i\vec{k}\vec{r}_n} = \underline{WA} = 1 \tag{13}$$

Minizing the expression $\underline{WRW}^H$ under the constraint $\underline{WA} = 1$ is performed using the Lagrangian operator and leads to (Capon et al., 1967, Capon, 1969)

$$\underline{W} = \frac{\underline{R}^{-1}\underline{A}}{\underline{A}^H\underline{R}^{-1}\underline{A}} \tag{14}$$

The "Capon estimator" is then

$$\underline{P}_{Capon} = \frac{1}{\underline{A}^H\underline{R}^{-1}\underline{A}} \tag{15}$$

The Capon estimator allows a higher angular resolution than the conventional estimator and the Rayleigh limit of the array is pushed away to lower wavenumber values allowing thus the characterization of waves propagating at close wavenumbers.

2.4 Multiple Signal Classification (MUSIC)

This method developed by Schmidt (1981, 1986a, 1986b) relies on the properties of the CSM matrix. This matrix is symmetric hermitian and can thus be decomposed as follows

$$\underline{R} = \underline{USU}^H + \underline{NN}^H \tag{16}$$

where

$$\underline{S} = \begin{bmatrix} |S_1(f)|^2 & \cdots & \cdots & 0 \\ \vdots & |S_2(f)|^2 & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & |S_q(f)|^2 \end{bmatrix}$$

and

$$\underline{U} = \begin{bmatrix} e^{i\vec{k}_1\vec{r}_1} & \dots & e^{i\vec{k}_q\vec{r}_1} \\ \vdots & \ddots & \vdots \\ e^{i\vec{k}_1\vec{r}_N} & \dots & e^{i\vec{k}_q\vec{r}_N} \end{bmatrix}$$

MUSIC uses the fact that the eigenstructure of $\underline{R}$ consists of a signal subspace spanned by the eigenvectors related to the q largest eigenvalues and a noise subspace related to the eigenvectors of the $N-q$ smallest eigenvalues. The singular value decomposition of $\underline{R}$ leads thus to

$$\underline{U} = \underline{E}_S \underline{\Lambda}_S \underline{E}_S^H + \underline{E}_N \underline{\Lambda}_N \underline{E}_N^H \tag{17}$$

where $\underline{E}_S$, $\underline{E}_N$, $\underline{\Lambda}_S$ and $\underline{\Lambda}_N$ are the eigenvectors and eigenvalues of the signal and noise subspaces, respectively. $\underline{E}_S$ and $\underline{E}_N$ are of dimension $N \times q$ and $N \times N-q$, whereas $\underline{\Lambda}_S$ and $\underline{\Lambda}_N$ are $q \times q$ and $N-q \times N-q$, respectively..

Because of the orthogonal property between signal and noise subspaces, the steering vectors of the signals are such that their projection into the noise subspace is minimum :

$$\underline{E}_N^H \underline{A} = \underline{0} \tag{18}$$

with $\underline{A}$ being the matrix formed from the steering vectors $\vec{a}(\vec{k}_i)$

$$\underline{A} = \begin{bmatrix} e^{i\vec{k}_1\vec{r}_1} & \dots & e^{i\vec{k}_q\vec{r}_1} \\ \vdots & \ddots & \vdots \\ e^{i\vec{k}_1\vec{r}_N} & \dots & e^{i\vec{k}_q\vec{r}_N} \end{bmatrix} = \begin{bmatrix} \vec{a}(\vec{k}_1)^T & \vec{a}(\vec{k}_r)^T & \dots & \vec{a}(\vec{k}_q^T) \end{bmatrix}$$

Steering vectors can thus be found by finding the peaks of the directional function (MUSIC spectrum)

$$\underline{D} = \frac{1}{\underline{A}^H \underline{E}_N \underline{E}_N^H \underline{A}} \tag{19}$$

or, equivalently

$$D(\vec{k}) = \frac{1}{\sum_{j=q+1}^{N} |\vec{a}(\vec{k}_j)^T \vec{e}_{Nj}|^2} \tag{20}$$

where $\vec{a}(\vec{k}_j)$ is the steering vector of the j-th signal and $\vec{e}_{Nj}$ the eigenvector of $\underline{E}_N$ related to the j-th eigenvalue.

The MUSIC algorithm relies on the property of the CSM matrix that can be decomposed in signal and noise subspaces. The orthogonality of signal and noise subspaces are exploited to find the roots of the signal propagation vectors (steering vectors) projected into the noise subspace. Given a reliable estimate of the CSM, MUSIC exhibits a greater angular resolution than Conventional's and Capon's algorithm However, this technique requires that the number of propagating waves be known or accurately estimated before decomposing the CSM into signal and noise subspaces. In case of stationary non-correlated waves, the estimation of the number of signals can be determined from the CSM matrix using the AIC or MDL criterion (Wax and Kailath, 1985). However, such criteria most generally fail when waves are correlated or when the noise is not spatially white.

2.5 MSPAC method

The spatial autocorrelation (SPAC) method was originally proposed by Aki (1957). This method is based on the assumption that the noise wavefield is stochastic and stationary in both time and space.

The spatial correlation function between signals recorded in the time interval $[0, T]$ at two receivers is given in the time domain by

$$\phi(r,\varphi) = \frac{1}{T}\int_0^T u(x,y,t)u(x + r\cos(\varphi), y + r\sin(\varphi), t)dt \tag{21}$$

where x, y and $x + r\cos(\varphi), y + r\sin(\varphi)$ are the Cartesian coordinates of the two receivers, r is the distance between receivers and φ denotes the azimuth of the direction between the two receivers. In case of a single dispersive wave propagating along the azimuth θ, Aki (1957) has shown, using the relation between the spectrum in time and the spectrum in frequency, that the correlation function can be expressed as

$$\phi(r,\varphi) = \frac{1}{\pi}\int_0^\infty \Phi(\omega)\cos\left[\frac{\omega r}{c(\omega)}\cos(\theta - \varphi)\right] d\omega \tag{22}$$

where $\Phi(\omega)$ is the cross spectrum, ω is the angular frequency and $c(\omega)$ is the velocity. Filtering now the wave through a narrow-band filter around the frequency ω_0, the cross spectrum can be expressed as

$$\Phi(\omega) = \Phi(\omega_0)\delta(\omega - \omega_0), \quad \omega > 0 \tag{23}$$

where $\delta(\omega - \omega_0)$ is the Dirac function. Hence, equation 21 becomes

$$\phi(r,\varphi,\omega_0) = \frac{1}{\pi}\Phi(\omega_0)\cos\left[\frac{\omega_0 r}{c(\omega_0)}\cos(\theta - \varphi)\right] \tag{24}$$

The correlation coefficient is then defined as

$$\rho(r,\varphi,\omega_0) = \frac{\phi(r,\varphi,\omega_0)}{\phi(0,\varphi,\omega_0)} \tag{25}$$

or simply:

$$\rho(r,\varphi,\omega_0) = \cos\left[\frac{\omega_0 r}{c(\omega_0)}\cos(\theta - \varphi)\right] \tag{26}$$

Equation 25 indicates that the correlation coefficient decreases more rapidly with increasing frequency along the propagation direction ($\varphi = \theta$). Although the graphical representation of

$\rho(r, \varphi, \omega_0)$ allows to give an estimate of the direction of propagation, in general, θ is not known and the azimuthal average of the correlation coefficient is introduced

$$\overline{\rho}(r, \omega_0) = \frac{1}{\pi} \int_0^{\pi} \rho((r, \varphi, \omega_0) d\varphi \tag{27}$$

and, using equation 24

$$\overline{\rho}(r, \omega_0) = J_0 \left(\frac{\omega_0 r}{c(\omega_0)} \right) \tag{28}$$

where J_0 is the zero-order Bessel function

$$J_0 = \frac{1}{\pi} \int_0^{\pi} \cos(x \cos(\varphi)) d\varphi \tag{29}$$

Equation 27 is available for non polarized waves, i.e. for Rayleigh waves recorded by vertical components.

The phase velocity $c(\omega_0)$ can thus be derived by matching the Bessel function J_0 to the azimuthal average of the correlation coefficients. These last are obtained by measuring $\rho(r, \varphi, \omega_0)$ for several stations evenly spaced around a semicircle of radius r with respect to a reference receiver at the center. Repeating this procedure as a funtion of ω yields to the estimation of $c(\omega)$. For best results in fitting the correlation coefficient, r has to be defined in such a way that the measured correlation coefficients span at least the first arch of the Bessel function J_0 over the frequency bandwith of interest. Ferrazzini et al. (1991) suggested to take r as the half-wavelength of the signal of interest.

Computation of correlation coefficients

The correlation coefficients can be measured in different manners
Computation of the normalized cross-spectra, then azimuthal averaging using the real part of the cross-spectra This is equivalent to the computation of the cross-spectra,and then computation of the correlation coefficients using the Fourier coefficients
Let us consider two signals $u_1(t)$ and $u_2(t)$

$$u_1(t) = \sum_n A_{1n}\cos(\omega_n t) + B_{1n}\sin(\omega_n t)$$

$$u_2(t) = \sum_n A_{2n}\cos(\omega_n t) + B_{2n}\sin(\omega_n t)$$

where A and B are the Fourier coefficients. The correlation function is:

$$\phi(r,\varphi,\omega_n) = E[u_1(t)\cdot u_2(t)]_t$$

thus:

$$\begin{aligned}\phi(r,\varphi,\omega_n) &= \tfrac{1}{T}\int_0^T \left[A_{1n}A_{2n}\cos^2(\omega_n t) + A_{1n}B_{2n}\cos(\omega_n t)\sin(\omega_n t)\right] dt + \\ &\quad \tfrac{1}{T}\int_0^T \left[B_{1n}A_{2n}\cos(\omega_n t)\sin(\omega_n t) + B_{1n}B_{2n}\sin^2(\omega_n t)\right] dt\end{aligned}$$

As $\int_0^T \cos^2(\omega_n t)dt = \int_0^T \sin^2(\omega_n t)dt = 1/2$ and $\int_0^T \cos(\omega_n t)\sin(\omega_n t)dt = 0$, the correlation coefficient is then determined by:

$$\phi(r,\varphi,\omega_n) = \frac{1}{2}\left[A_{1n}A_{2n} + B_{1n}B_{2n}\right]$$

$$\phi(r,\varphi,\omega_n) = \frac{1}{2}\sqrt{\left[(A_{1n}^2 + B_{1n}^2)(A_{2n}^2 + B_{2n}^2)\right]}$$

In the current implementation, we first pre-filter all signals by applying a cosine taper of certain bandwidth to the Fourier coefficients of the signal spectra and backtransforming again into time domain. Then the correlation to zero lag for all station pairs are obtained by computing the cross correlation coefficient in time domain. Finally the crosscorrelation coefficients are averaged as suggested by Bettig et al. (2003).

When the arrays do not have a perfectly circular shape, one can no longer estimate the azimuthal averaged correlation coefficients at constant radius. Bettig et al. (2003) have thus introduced an additional integration over rings $r_1 \le r \le r_2$ of finite thickness as follows

$$\begin{aligned}\overline{\rho}(r_1,r_2,\omega_0) &= \tfrac{1}{\pi}\tfrac{2}{r_2^2-r_1^2}\int_0^\pi\int_{r_1}^{r_2}\rho(r,\varphi,\omega_0)r\,dr\,d\varphi \\ &= \tfrac{1}{\pi}\tfrac{2}{r_2^2-r_1^2}\int_0^\pi\int_{r_1}^{r_2}\cos(\tfrac{r\omega_0}{c(\omega_0)}\cos(\theta-\varphi))r\,dr\,d\varphi\end{aligned} \quad (30)$$

Using the first-order Bessel function $J_1(x) = \int xJ_0(x)dx$, equation 30 becomes

$$\begin{aligned} \overline{\rho}(r_1, r_2, \omega_0) &= \frac{2}{r_2^2 - r_1^2} \int_{r_1}^{r_2} rJ_0(\frac{r\omega_0}{c(\omega_0)})dr \\ &= \frac{2}{r_2^2 - r_1^2} \frac{r\omega_0}{c(\omega_0)} \left[rJ_1(\frac{r\omega_0}{c(\omega_0)}) \right]_{r_1}^{r_2} \end{aligned} \tag{31}$$

An array can thus be divided into several equivalent semicircular sub-arrays k defined by the sensor pairs (i, j) such as $r_{k_1} \leq r_{ij} \leq r_{k_2}$. The average correlation coefficient is then

$$\overline{\rho}(r_k, \omega_0) = \frac{1}{\pi} \frac{2}{r_{k_2}^2 - r_{k_1}^2} \sum_{r_{k_1} \leq r_{ij} \leq r_{k_2}} \rho(r_{ij}, \varphi_{ij}, \omega_0) \overline{r}_k \Delta r_k \Delta \varphi_i j \tag{32}$$

where

$$\overline{r}_k = \frac{r_{k_1} + r_{k_2}}{2}$$

$$\Delta r_k = r_{k_2} - r_{k_1}$$

$$\Delta \varphi_{ij} = \frac{\varphi_{ij+1} - \varphi_{ij-1}}{2}, \qquad \sum_{r_{k_1} \leq r_{ij} \leq r_{k_2}} \Delta \varphi_i j = \pi$$

The determination of r_{k_1} and r_{k_2} is a compromise between the number of sensors per ring which should fix the azimuthal distribution and the ratio $\Delta r_k / \overline{r}_k$, which should be as small as possible.

The algorithm used for fitting the correlation coefficients to the Bessel functions in order to retrieve $c(\omega)$ is a nonlinear inversion based on least square adjustment (Tarantola and Valette., 1982). The equation is considered as a non linear relation of the form $\vec{d} = g(\vec{m}$, where $\vec{d}$ is the data vector (here the correlation coefficients) and $\vec{m}$ the model vector (here the frequency dependent phase velocities). The least square problem is solved by using the iterative algorithm

$$\vec{m}_{i+1} = \vec{m}_0 + \underline{C}_{m_0 m_0} \underline{G}_i^T \left(\underline{C}_{d_0 d_0} + \underline{G}_i \underline{C}_{m_0 m_0} \underline{G}_i^T \right)^{-1} \left[\vec{d}_0 - g(\vec{m}_i) + \underline{G}_i (\vec{m}_i - \vec{m}_0) \right] \tag{33}$$

where i is the iteration index, $\vec{m}_0$ is the a priori model vector, $\underline{C}_{d_0 d_0}$ and $\underline{C}_{m_0 m_0}$ are the covariance matrices for data and model vectors, respectively. $\underline{G}_i$ is the partial derivative matrix $G^{kl} = \partial g^k / \partial m^l$ and $\underline{G}_i^T$ its transpose. The estimation of the errors on the parameters are obtained from the a posteriori covariance matrix

$$\underline{C}_{mm} = \underline{C}_{m_0 m_0} - \underline{C}_{m_0 m_0} \underline{G}^T \left(\underline{C}_{d_0 d_0} + \underline{G} \underline{C}_{m_0 m_0} \underline{G}^T \right)^{-1} \underline{G} \underline{C}_{m_0 m_0} \tag{34}$$

3 Database connectivity

Array measurements of ambient vibrations are usually short term experiments. The typical duration of recording lies in the order of tens of minutes to a few hours. Nevertheless, the data amount to be handled in array analysis is relatively high depending on sampling rates and number of channels involved. Besides the raw waveform data and corresponding timing information it is necessary to keep track of station coordinates and instrument information (both sensor and datalogger settings information). This is especially an issue when several array configurations have been deployed at the same site. In order to deal efficiently with this information during data processing a data base approach comes handy.

3.1 CAP and GIANT

GIANT has been developed by A. Rietbrock (Rietbrock and Scherbaum, 1998) for the consistent analysis of large data sets of local/regional earthquake waveform data. It has been extensively used in large temporary network deployments and the analysis of heterogeneous aftershock data sets. Its base is a database structure designed for holding both logistic background information of station networks (locations, calibration of instruments, start/stop times) as well as analysis results from phase picking, hypocenter and focal mechanism solutions together with the velocity model used for obtaining the solutions and spectral fitting results. The waveform data and calibration data itself are not stored directly, but referenced by format type and relative location in the file system. The use of environment variables allows to change the location of files within the file system or the access from direct access archiving media like CDROM or DVD.

Using a graphical user interface the data base can be queried for different data sets (waveforms, phase information, calibration data, station/hyopcenter locations, focal mechanisms, etc.) and query results are visualized or passed to external seismological analysis software for interactive or batch analysis. Altough the original design was specialized for event-based time chunks of waveform data, GIANT has been used now for years also for the analysis of continuous data sets from short and long term seismological experiments.

Being involved in the GIANT development on the users's side since several years (testing, documentation and writing batch query applications) it has been a natural step for me to use GIANT for the data management of ambient vibration array measurment campaigns. For the purpose of **CAP** , just a small part of the data base structure is actually used. The information stored into GIANT regards the station positions, instrument calibration data and the location of the waveform data on the disk.

Running **CAP** with the GIANT interface requires an existing GIANT database structure and a set of environment variables pointing to the location of data base files and the base directories of waveform and calibration data. for the setup of a GIANT data base, the user is referred to the GIANT manual. A pdf version of this document can be downloaded from software page of the Institute of Geosciences, University of Potsdam, GIANT download page or direclty from this link http://www.geo.uni-potsdam.de/Forschung/Software/downloads/giant.pdf

3.2 CAP and GEOPSY (former SARDINE)

During the development of **CAP** within the SESAME community, it became obvious, that most future users of this software would like a less "unix-like" version of the software. However, for the reasons given above there is still the need for an underlying database structure. Fortunately Marc Wathelet offered a solution with his database development "GEOPSY" (formerly called SARDINE).

Similar to GIANT, GEOPSY has also been developed in its beginnings for a different purpose (shallow seismic refraction surveys). Nonetheless, the necessary information within the context of handling data from ambient vibration array experiments (geometry and waveform data) could be used from the very beginning with GEOPSY (then still called SARDINE). GEOPSY stores the database information in a single ASCII format file. The file format is easy, but proprietary. GEOPSY accesses information from the database with a **Qt**-based GUI-interface. The use of **Qt** (©Trolltech), as well as the storage of the database information into a simple ASCII file format makes GEOPSY a really portable software package. Until now, GEOPSY has been tested on Linux, Windows and MacOS-X operating systems (and connected with that on quite some different hardware platforms).

CAP with the GEOPSY interface can also be run from the command line. The option flags for the command line are extended by the "-d" flag, which takes a GEOPSY database name as argument.

3.3 CAP and FAKE_DB

In a very recent development **CAP** has been improved for the purists among us. For those who just want to try out the software package or just deal with small data sets, it is now possible to use **CAP** without the need of creating a database beforehand (neither GIANT nor GEOPSY).

The necessary information of the array setup, like station names, station coordinates, station calibration and waveform data file names are read from two separate ASCII files specified at the command line. It is probably the simplest way to get **CAP** running out of the box.

In this version, **CAP** reads the information provided in the ASCII files from the command line and creates database structures in memory only (on basis of the internal data presentation of GIANT) and then just proceeds processing. Specification of the ASCII file formats used are given in section 7.1

4 Preprocessing Block

4.1 Integer Decimation

For microtremor wavefields it is known that the spatial coherency of signal arrivals is rather limited due to the low signal to noise ratio of the analysed time window. Indeed it is difficult to talk about signal to noise ratio within the scope of microtremor analysis. Any part of the wavefield is considered as signal which contains information about the propagation properties of the structure, but it is not so clear which part of the wavefield we would classify as "noise" in the usual sense (compare discussion in Scales and Snieder, 1998). In this context we could refer to signal as coherent plane wave arrivals whereas "all other wave arrivals" in the array are considered noise.

The fact, that the spatial coherency lengths are usually small makes it necessary to choose relatively small array apertures and inter-station distances in order to ensure coherent signal arrivals at all stations within the array and further to reduce effects of curved wavefronts of nearby sources. Consequently, small inter-station distances result in small travel times for plane wave arrivals at the array sensors and it is therefore common practice in microtremor array measurement experiments to use rather high sampling rates to ensure a good time resolution.

In case the frequency band of interest for the analysis is set to very low frequencies if compared to the original sample rate, it might be of interest to downsample the raw data streams to reduce the computational load for the analysis. For these (rather rare) situtations **CAP** provides a simple integer decimation option to reduce the sampling rate.

The keyword for the decimation option in the configuration file is called "DECIMATE" and the values expected are of type integer. Any value less than 2 turns the decimation off, any value larger or equal than 2 is interpreted as the decimation factor for downsampling.

4.2 Butterworth Bandpass Filtering

This option has been kept for historic reasons mainly. The need for applying a bandpass filter to the data in the context of dispersion curve determination is rather limited.

The keyword used for selecting a Butterworth bandpass filter is called "BBP_FILTER". The value is of type integer and can be either 0 or 1. Setting "BBP_FILTER" to 0 deselects bandpass filtering in the pre-cprocessing stage, whereas the value 1 enables the filtering.

If bandpass filtering is selected, the filter parameters are specified via the keywords: "BBP_LOW", "BBP_HIGH", "BBP_ORDER" and "ZERO_PHASE". BBP_LOW and BBP_HIGH require a float parameter and specify the lower and upper corner frequencies of the bandpass filter. BBP_ORDER and ZERO_PHASEi expect an integer value parameter. The value given for BBP_ORDER sepcifies the number of sections (number of conjugate complex pole pairs) used for the filter design. The allowed value range for this parameter is 1 to 9. Each section adds another 6dB/decade roll-off to the flanks of the filter.

The value for the ZERO_Phase flag is a toggle option and can be either 0 or 1. "0" toggles this option off, whereas "1" designs the Butterworth bandpass filter to be zero-phase. The zero-phase properties of filter are realized by forward-backward filtering of the time series, thus the number of sections specified with the BBP_ORDER value are effectively doubled. Thus each section accounts then for a 12dB/decade roll-off at the filter flanks.

4.3 Instrument simulation

The *instrument simulation* feature implemented in **CAP** relates to the necessity of dealing with heterogenous recording equipment in real-world array experiments. Especially important is the correction of the phase delays caused by the recording instrument (compare SESAME-Deliverable D05.07)

The keyword for selecting the instrument simulation option is called "SEIDL", as the algorithm for instrument simulation has been proposed by Seidl (1980). SEIDL takes an integer argument, which is either 0 or 1. "0" toggles this option off, "1" selects the simulation of a common instrument response for all selected sensors. The parameters for the simulated instrumet response are given by the keywords "FSIM" and "HSIM". Both keywords require a float parameter. FSIM denotes the corner frequency of the simluated response and HSIM is the parameter of critical damping in the range from $[0., 1.]$.

If the instrument simulation option is selected, **CAP** reads the calibration information provided as GSE1 Pole and Zero file format for each channel which is to be processed. It determines then a simulation filter composed of the inverse frequency response of the recording sensor and the desired reponse determined from the parameters specified via the FSIM and HSIM keywords. The original waveform data in the database is not changed.

*Note: This option has only effect in the GIANT_DB and FAKE_DB versions of **CAP** , but not with GEOPSY interface. Setting this option with GEOPSY version of **CAP** will return an error message and stop processing.*

4.4 Additional processing settings

There are two additional processing settings which have to be mentioned in the realm of the preprocessing stage of **CAP** . These options are probably rarely used, but have been necessary for special datasets during the course of SESAME.

Using the keyword GAUSSNOISE it is possible to add gaussian noise to the waveforms *before* processing. The value is of type float and specifies the standard deviation of the random samples to be added. It must be noted, that this value is not to be understood as absolute standard deviation, but as a factor multiplied to the standard deviation determined from the individual traces before adding the gaussian samples. Selecting for example a value of 0.1 translates therefore to a standard deviation of $0.1 * \sigma_i$ where i is the trace (station index). Setting this value to any negative number disables this option.

If some timing error has occurred during the data acquisition, we implemented an option to account at least for known static time shifts for individual records. The keyword to use this

feature is called TIME_CORR and is used as a switch variable. A value of 1 activates this functionality, choosing 0 turns it off. If time correction is chosen, an *user-interaction is required.* During the preprocessing stage the user will be queried which stations shall be corrected for timing. At this query the user has to specify a string of station names separated by plus signs (no white spaces or other delimiters are allowed here). Then the user will be asked to enter the time delays (in number of samples) for each station. Delayed records must be given a negative value, "early" records must be specified by positive sample values. As the routine only shifts the pointers on the records by an integer number of samples and does not correct for time delays of fractions of samples, we recommend to correct the records *before* building a database and using **CAP** . However, this option has been an easy solution for a sporadic occurrence of missing time corrections for leap seconds since 1970 due to a bug in the GPS-card in our case.

5 Main Processing Block

In this section we give a short overview of the different methods implemented in **CAP** , their options and parameters and how to select the correct processing parameters. The majority of subsections are connected with the usage of different frequency wavenumber techniques, followed by a subsection dealing with the SPAC method and finally complemented by some experimental method implementations.

As all (except of one) f-k methods are related to a grid search for the determinion of the propagation characteristics of the seismic wavefield in time and frequency, we begin with a general description of time-frequency tiling and wavenumber grid layout in **CAP** .

5.1 Determination of time-frequency tiling

For the desired goal of the determination of phase velocity curves $c(\omega)$ from continuously recorded microtremor array data, the array analysis has to be performed within a set of narrow frequency bands. Thus, the user has to specify, how the frequency bands should be constructed. The selection of individual time window lengths of data chunks for processing should be usually adapted to the frequency band under consideration.

In **CAP** the time frequency tiling can be controlled by the use of two sets of keywords. For the specification of the frequency bands the following keywords are used: "NUM_BANDS", "LOWEST_CFREQ", "HIGHEST_CFREQ", "BANDWIDTH" and "BANDSTEP", whereas the keywords "WINFAC", "OVERLAP", "WINLEN" and "STEP" are available for the selection of time windows.

- Frequency tiling:

  NUM_BANDS gives the number of frequency bands to be used for the analysis. The bandwidth of each frequency band is controlled by BANDWIDTH, which gives the half-bandwidth as fraction of the center frequency, thus the frequency band is limited from $f^n_{low} = (1 - BW)f^n_c$ to $f^n_{high} = (1 + BW)f^n_c$. The central frequencies for each band are selected to be spaced equidistantly on a logarithmic frequency band. The keyword LOWEST_CFREQ specifies always the center frequency for the lowest frequency band. The value given for the keyword HIGHEST_CFREQ is used only, if the keyword BANDSTEP has given a value less than zero. In this case NUM_BANDS frequency bands are spaced between LOWEST_CFREQ and HIGHEST_FREQ and the successive step between center frequencies is selected automatically. Using values larger than one for BANDSTEP, NUM_BANDS frequency bands with the chosen BANDSTEP according to $f_c^{n+1} = BSf_c^n$. In this case the value for HIGHEST_CFREQ is ignored and the highest center frequency is then $f_c^N = f_c^0 BS^{N-1}$, where N is the value given for NUM_BANDS.

- Specification of analysis window:

  For the specification of sliding analysis window parameters, two options are available. First, using keywords WINLEN and STEP, constant length time windows of length WINLEN

seconds are used for all frequency bands under consideration. Successive analysis windows are shifted by STEP seconds. This option necessarily means, that the time-bandwidth product of analysis windows changes from frequency band to frequency band. This way of anaylsis window specification can be turned off by the use of the keyword WINFAC. If WINFAC is set to values larger than zero, both the WINLEN and the STEP keyword are always ignored. The window length is then chosen as WF/f_c^n (WF being the value of WINFAC), whereas the step between successive time windows is controlled by the keyword OVERLAP. OVERLAP can be set to negative values or must lie in the range of $[0, 1]$. negative values select the overlap of succesive time windows to 50% for all frequency bands. The time step in seconds is therefore $0.5WF/f_c^n$.

In the following figures 1 to 5 some examples of the usage of the above discussed parameters are shown. For the time-frequency tiling given in Fig. 1 the following settings have been used:

NUM_BANDS	15
LOWEST_CFREQ	0.5
HIGHEST_CFREQ	1.5
BANDSTEP	-1
BANDWIDTH	0.1
WINFAC	10
OVERLAP	-1

In the left panel, the frequency axis is linear, while in the right the the frequency axis is displayed logarithmically. The first time window for each frequency band is indicated by gray colors to better differentiate the individual time windows shifted by 50% of the window length.

The above settings are probably the most common selection for the purpose of analysing ambient vibration data. It should be noted, however, that the time windows in individual frequency bands are not aligned to a specific time base and the number of time windows processed increases for higher center frequencies.

For the second example shown in Fig. 2 the OVERLAP parameter has been changed to achieve a constant time shift over frequency bands. In the left panel OVERLAP has been set to 1 (50% overlap of window length for highest frequency band) whereas for the right example OVERLAP was set to 0 (50% overlap of window length for the lowest frequency band). Note, that in the left the lower frequency bands are highly oversampled in time leading to a large total number of windows for processing which causes long computation times. On the contrary in the opposed example, the total number of windows selected for processing is low but part of the data for higher frequencies is not evaluated at all.

A compromise between the extreme settings of the parameter OVERLAP in the previous example is shown in Fig. 3. Here OVERLAP ist set to 0.7. Now, the number of time windows is acceptable and no gaps occurr for any frequency band under consideration whereas the time shift is still constant for all frequencies.

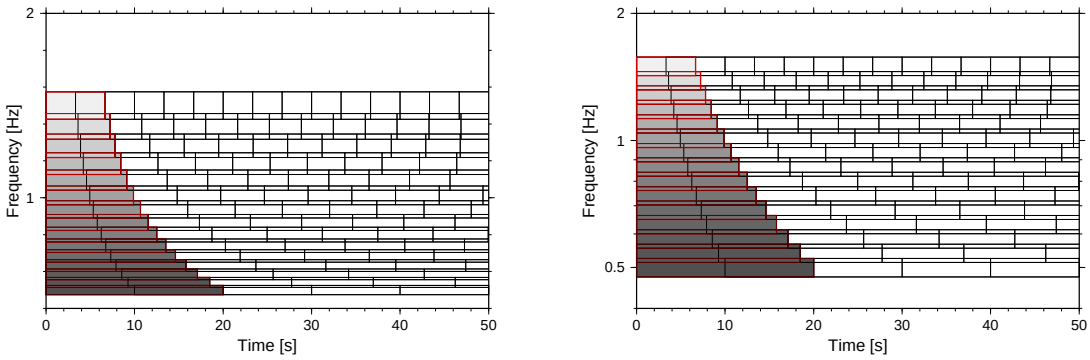

Figure 1: Example of tiling in time-frequency plane. The settings, as given in Table 5.1, are a common choice for processing ambient noise wavefields for the determination of dispersion characteristics

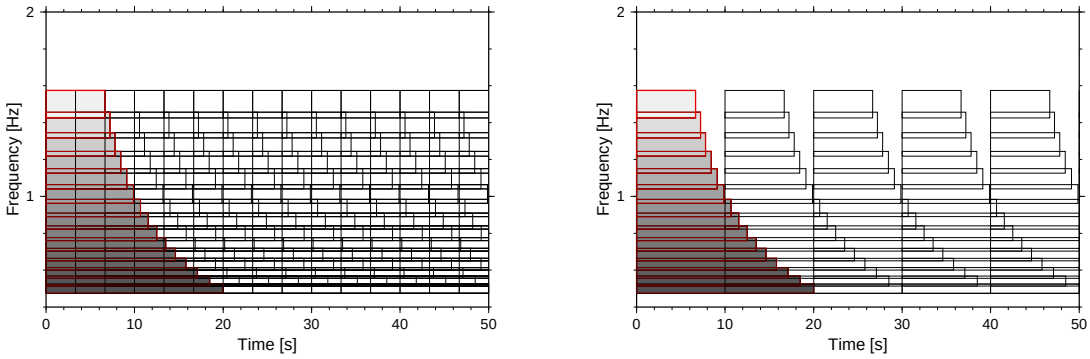

Figure 2: Example of tiling in time-frequency plane. A constant time shift between successive time windows is shown, whereas the amount of time shift is bound to 50% of the highest (left panel) or lowest (right panel) center frequency

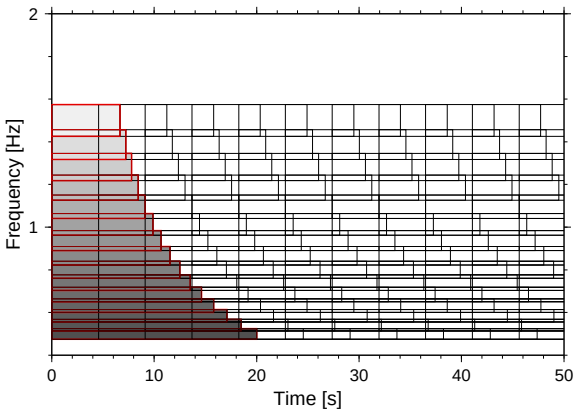

Figure 3: Example of tiling in time-frequency plane with intermediate constant time shift between successive time windows (time shift bound to an intermediate center frequency).

The examples given in Fig. 4 show results when the parameter BANDSTEP is set to a positive value. In this case the value given for HIGHEST_CFREQ is ignored and the center frequencies are selected as explained above. In the left panel BANDSTEP has been set to 1.15 whereas the parameter BANDWIDTH is 0.1. Thus the frequency bands don't overlap. In the example to the right BANDSTEP is 1.05 which results in highly overlapping frequency bands. In both cases the parameter OVERLAP was set to 1 resulting in a constant time shift which resembles 50% of the highest frequency band.

In the last example we have used

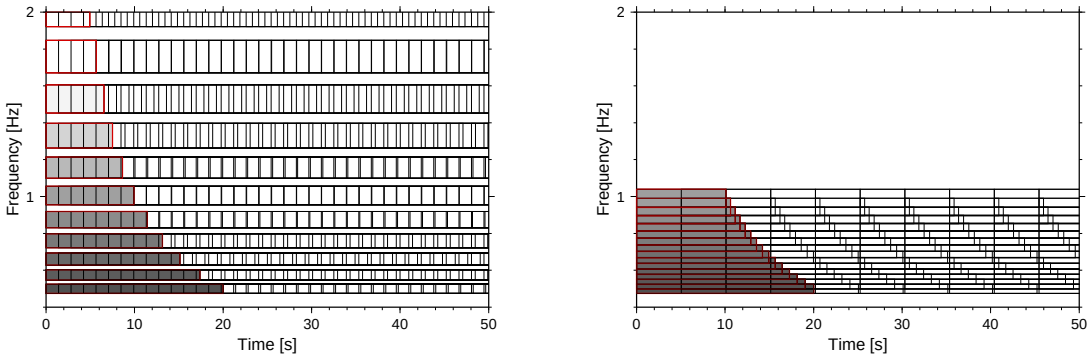

Figure 4: Example of tiling in time-frequency plane demonstrating the influence of the BANDSTEP parameter. Left: large BANDSTEP value. Right: small BANDSTEP value.

Within the program flow of **CAP** an internal list of time frequency cells is computed from the settings specified in the configuration file. The advantage of doing so is two-fold. On one hand this procedure enables an efficient looping over frequency bands and time windows within

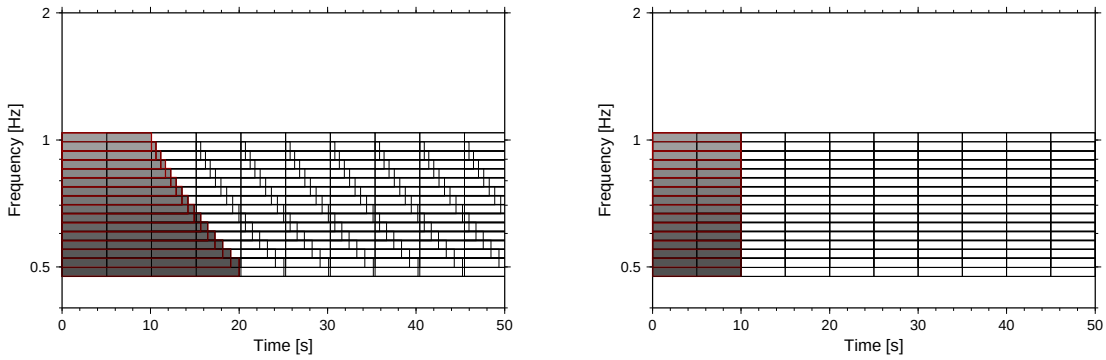

Figure 5: Example of tiling in time-frequency plane. Left: window length inversly related to center frequency. Right: constant time windows for all frequencies.

each of these bands without the necessity to re-code the computations for each method (spares significant amount of code lines). More important is the possibility of keeping track of processed data chunks by storing the determined time frequency information into a file and the re-import of a properly formated file to enable arbitrary time-frequency processing schemes. This especially enables the usage for any pre-processing scheme which tests the apropriateness of individual time-frequency cells for processing and to exclude bad or inapropriate data from processing in an easy and comfortable way (see for example section 5.9.1).

5.2 Determination of f-k grid layout

Typically frequency wavenumber methods are linked to a grid search over the wavenumber domain in order to obtain the signal parameters of the most powerful or most coherent signal crossing the array. How well the signal parameters are estimated does also considerably depend on the sampling employed for the grid search. Within **CAP** , the following schemes of wavenumber sampling are implemented:

- polar grid layout with equidistant spacing in r and ϕ directions, where the radial coordinate is either proportional to apparent velocity or slowness (not wavenumber).
- cartesian grid layout with equidistant spacing in x and y directions, either proportional to apparent velocity or slowness (not wavenumber).
- linear grid layout with equidistant spacing in x direction, either proportional to apparent velocity or slowness (not wavenumber).

The keyword used to switch the layout of the wavenumber grid (or line) is called GRID_LAYOUT. GRID_LAYOUT can be set to $0, 1$, or 2, indicating polar, cartesian or linear grid sampling. For the linear "grid", an additional parameter is needed which specifies the direction from the array center to the source or the direction, on which the wavenumber evaluation should be projected.

The keyword for this parameter is called LINEAR_PHI and takes values in the range $[0, 360]$ indicating the backazimuth (angle measured from N over E when loooking from the array center to the source).

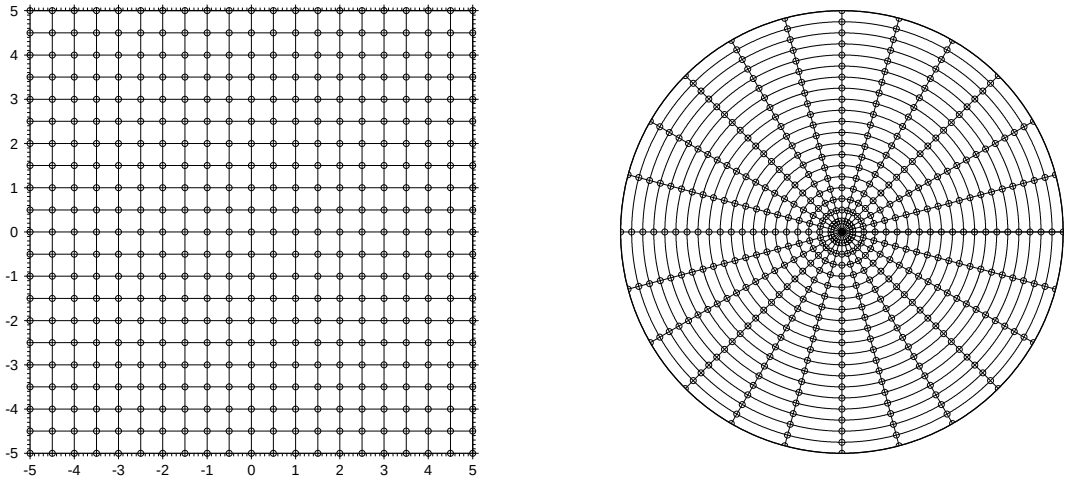

Figure 6: Possible layout of wavenumber grids. Left: cartesian grid, right: polar grid. Linear grid layout is not shown.

As indicated above, **CAP** supports both equidistant sampling in slowness as well as for apparent velocity. The keyword GRID_TYPE is used to toggle between both options. The argument value is expected as integer with 0 indicating a layout sampling linearly in slowness and 1 chooses the apparent velocity grid. It is recommended to use the sampling with equidistant spacing proportional to slowness as it is more appropriate in terms of error analysis. The measurement error (travel time delays) is linearly proportional to slowness but inversely related to apparent velocity.

The maximum value of the grid is given by the keyword GRID_MAX. It is always specified as float value in the unit of the selected GRID_TYPE. For example, a value of 5.5 means either $5.5 s/km$ for a slowness proportional layout or $5.5 km/s$ otherwise.

Finally, the grid dimensions are specified by the keyword GRID_RESOL and in case of choosing a polar grid layout additionally the keyword NPHI must be given. Both keywords expect an integer value as argument. In case of a cartesian grid layout (GRID_LAYOUT equals 1) the sampling invterval is determined as 2*GRID_MAX/(GRID_RESOL-1), whereas for a polar grid the radial axis is sampled in intervals of GRID_MAX/(GRID_RESOL-1). The azimuthal resolution for polar layouts is 2π/NPHI. For the linear grid layout, the slowenss/apparent velocity resolution along the chosen direction is given by GRID_MAX/(GRID_RESOL-1).

It should be noted that the selection of the grid dimensions (GRID_RESOL and/or NPHI) is crucial for the number of computations which have to be performed and therefore controls the overall speed of processing. Furthermore, the necessary storage amount for the output of f-k

grids increases linearly with the grid dimensions and must therefore be considered in case of limited available disc space.

5.3 Conventional frequency wavenumber analysis – CVFK

The conventional frequency wavenumber approach has been implemented in three different ways into **CAP** . We distinguish between a semblance based approach after Kvaerna and Ringdahl (1986) (CVFK), the power-based evaluation of the slowness map via the averaged cross spectral matrix (CVFK2) and a gridless approach (CVFK_FAST) which tries to find the maximum in slowness maps by a non-linear optimization technique. The usage of these methods and related processing settings are discussed in the following

5.3.1 Semblance based estimates for individual time windows - CVFK

The CVFK method implemented in **CAP** is selected by setting the keyword METHOD to 0. The CVFK computes for each individual time-frequency cell a complete slowness map. The time-freuency tiling and the setup of the slowness grid dimensions and resolution are specified as explained in sections 5.1 and 5.2. For each grid point in the slowness map the following expression is evaluated:

$$RP(\omega_c, \vec{s}) = \frac{\sum_{k=1}^{K} |\sum_{n=1}^{N} X_n(\omega_k) e^{i\omega_k \tau(\vec{s},\vec{r}_n)}|^2}{N \sum_{k=1}^{K} \sum_{n=1}^{N} |X_n(\omega_k) e^{i\omega_k \tau(\vec{s},\vec{r}_n)}|^2} \tag{35}$$

$X(\omega_k)$ are the Fourier coefficients at discrete frequencies ω_k computed from the waveforms recorded at station n located at position $\vec{r}_n$. The delay times $\tau(\vec{s},\vec{r}_n) = \vec{s}\vec{r}_n = s_x r_{xn} + s_y r_{yn}$ account for the travel times to stations n for a plane wave propagating across the array with slowness vector $\vec{s}$. The double sum is evaluated over N stations and K discrete frequencies. The frequency limits are given from the selection of time-frequency cells. *Note: choosing small bandwidths and short time windows may cause a situation where no discrete frequencies ly within the frequency band of interest!.*

The value RP can be interpreted as an approximated semblance value (Neidell and Taner, 1971) and has been termed relative power in Kvaerna and Ringdahl (1986). In a grid search the maximum of RP is found and the parameters of plane wave propagation (absolute slowness and backazimuth) are then computed from the slowness vector maximizing RP. These values are recorded into an ASCII file together with the beampower value (normalization constant in equation above) for each of the processed time-frequency cells. Dispersion curve estimates are then obtained by obtaining the distributional characteristics from the ensemble of wavefield propagation estimates from this output file. This post-processing step is performed outside **CAP** with a small utility program called fk2disp (see section 6).

As the number of processed time windows is usually high (in the order of several thousands) it is not wise to store the individual slowness maps, as it would rapidly fill the available disk space. Instead, an averaged slowness map is computed for each frequency band and stored into

an ASCII file format. Furthermore (as for the later following f-k methods) a certain fraction of highest values from the slowness maps are stored into an ASCII file. The fractional amount of values stored is determined from the value (range $[0., 1]$) given for the keyword MAPFRAC. *It is highly recommended to use a very small value of MAPFRAC for the CVFK method (e.g. below* 0.001*) in order to avoid huge output files.*

The processing can be applied to individual single components by setting the keyword COMP to 1 (vertical), 2 (north) and 3 (east). Most appropriate for the goal of deriving disperion curves of the Rayleigh wave part of the ambient noise wavefield is to choose the vertical component. A more reasonable horizontal component processing is possible by choosing values 22 or 33 for the keyword COMP. The value of 22 is associated with the radial component of the wavefield, 33 with the tangential component. Both radial and tangential components are constructed from the original components of motion (north and east) by rotating the coordinate system into the direction of the actual tested slowness vector, that is, individually for each slowness grid point. As a result, the processing of radial or tangential wavefield components is very time consuming.

Until now, no full three component processing is implemented for the f-k methods.

5.3.2 Averaged cross spectral matrix approach - CVFK2

Opposed to the implementation described above, the second approach used for estimating dispersion characteristics of the wavefield by means of a conventional wavenumber algorithm does not compute slowness maps for each individual time frequency cell. We named this method CVFK2 and it is selected by specifying the value 1 for the keyword METHOD. The CVFK2 is based on the evaluation of the averaged cross spectral matrix (estimator P_{CV} as described in section 2.2). The cross spectral matrix is obtained from stacking the covariance matrices from individual time windows for each frequency band. After stacking a "sensor normalization" is aaplied and a single slowness map is computed for this frequency band. In this case the slowness maps show the distibution of beampower values associated with each slowness grid point.

The slowness maps are analysed to find the three highest local power maxima within the grid. The propagation parameters are computed from the locations of these maxima and are stored into an ASCII output file together with the associated beampower estimates. The full slowness maps for each frequency are additionally written to another ASCII file as well as the fraction of highest beampower values and associated propagation parameters (see above for the use of the keyword MAPFRAC). The evaluation of the "best" beampower values provides a means to give some uncertainty estimate for the CVFK2 computations.

As for the CVFK implementation, the CVFK2 algorithm can be applied to both single components as well as to the radial and tagential components of the ambient noise wavefield - no full three component anaylsis can be performed so far.

5.3.3 Semblance based estimates for individual time windows - gridless computation - CVFK_FAST

With the SESAME project it turned out, that the CVFK computation allows a robust determination of dispersion curves, as it is a numerically very stable algorithm. The problems of insufficient resolution for multiple plane wave arrivals at individual time steps can be conquered efficiently by looking on the long-term statistical distribution of estimates and not to rely on single wave propagation prameters. Unfortunately, the overall processing time tends to be very long, as for each individual time window a full slowness map has to be computed. Therefore we have implemented another approach in order to make the CVFK computations feasible for longer time series.

We use a simplex-simmulated annealing technique as described in Press et al. (1992) to find the maximum of the semblance function (equation 35) in a limited region of the slowness space. The use of this procedure is not undebated among the members of the SESAME group. Due to the nature of non-linear optimization procedures (e.g. Sambridge and Moosegard, 2002) it is clear that the final estimate may correspond to a local maximum in the slowness map and then differs from the estimate obtained for a complete grid search in the wavenumber domain. However, we think that the following arguments make worth it a try:

- for a typical number of sensors used for ambient virbation array measurements (less than 10) and the narrow-band evaluation of the wavefields the array responses are suffciently smooth to allow the application of this non-linear optimization technique.
- the optimization is re-started several times from starting models covering distinct regions of the slowness maps. By doing so, we hope to avoid that a) some parts of the slowness maps are not sampled at all and b) in case that in a previous run only a loca maxima has been found, the restarting from different starting samples could drive the solution to the global maximum.
- test runs on both synthetic and real data sets show, that the distributions of wave propagation parameters resemble one another very closely. (see section 7.3).

Besides the legitimate criticism we observed the following advantages when using this implementation of CVFK. The typical saving of computation time is very high. It is linearly related to the number of function evaluations needed for convergence in the optimised search when compared to the full grid search. As typical grid sizes are of the order of 10000 or even larger, the number of computations involved for the simplex-simmulated search is (even restarting several times from different start samples) is seldomly more than 1000. Speedup of the processing times of factor 10 to 20 are therefore easily achieved. An additional advantage lies in the fact that the computations are no longer fixed to a certain grid spacing, thus resolution limits of slowness or apparent velocity estimate do no longer change for certain regions of the wavenumber space when sampled in one or the other domain (slowness or apparent velocity).

The CVFK_FAST method is selected by setting the integer value for keyword METHOD to a value of 14. Please note, that for the moment, only single component processing is implemented.

Thus, selections of 1, 2, or 3 for the keyword COMP is mandatory, 22, 33 or 123 are not allowed. The output of CVFK_FAST follows the same formatting as for the CVFK implementation.

5.4 Slantstack Analysis – SLANTSTACK

This option has been removed after the first part of the SESAME project. It had been implemented according to Louie (2000) and a roll-along experiment was conducted in order to test the performance of ambient vibration analysis for linear array layouts (Ohrnberger et al., 2001). The results of this test showed, that the projection of the obtained slowness estimates on a single look direction with unknown true source direction creates an unknwon amount of bias which can not be resolved. However, processing of linear array layouts can still be performed for active source experiments using the linear grid layout with all grid based f-k methods implemented.

5.5 Capon's high-resolution frequency wavenumber analysis – CAPON

The CAPON estimator, as described in section 2.3, is selected by setting the value for the keyword METHOD to 2. The algorithm is implemented as presented in Capon (1969), using the spectral crosscorrelation matrix which is equivalent to the cross sepctral matrix (CSM) besides a normlization which depends on the autopowers of the individual stations. This has been termed "sensor normalization" by Capon (1969) and improves significantly the results. One main advantage of this normalization technique is the numerical stability in the inversion of the cross spectral matrix, as the number range of the matrix is limited between $[0., 1.]$.

The cross spectral estimate is performed for each frequency band on the list of time windows computed previously from the user selected time-frequency tiling. After the averaging of the CSM, the f-k map is computed for the actual frequency band and the following results are stored into the output files:

- The location of the 3 highest maxima in the f-k map for each frequency.
- The complete f-k map
- The N highest power values of the f-k map where N is a fraction of grid points in the f-k map determined from the value given for MAPFRAC.

The CAPON estimator can be applied to any individual component selection which is achieved by setting the COMP paprameter to 1, 2 or 3 indicating Z, N, or E components. Furthermore, as for the CVFK (CVFK2) methods, selecting COMP as 22 or 33, acts on the radial or tangential components. The components are to be interpreted with respect to the propagation direction for each individual slowness vector tested in the grid search. Both horizontal components (N and E) must be available in the data base and radial or tangential components are computed for each point in the slowness grid. Until now, there is no full three-component processing available

For the specific format of CAPON output, please see section 7.2.

5.6 MUltiple SIgnal Classification – MUSIC

To apply the MUSIC method for ambient vibration array processing, the keyword METHOD must be set to either 5 or 6 (MUSIC or MUSIC2). Both algorithms are implemented in the same way, but differ in the estimation of the cross spectral matrix (CSM). For option 5, the MUSIC algorithm acts on CSM estimated from single time windows, whereas for option 6 (MUSIC2), a block-averaged CSM estimate equivalent to the implementation for the CVFK2 or CAPON is used.

As stated in section 2.4, the main advantage of the MUSIC algorithm is that it has higher resolution than the conventional FK for the separation of multiple sources interfering in the same time-frequency cell as well as for the estimation of their properties. Therefore, the optimal utilization of this algorithm needs to know the order of the model (i.e. q, the number of eigenvectors) that one has to use to describe the signal subspace.

The keyword for the selection of the number of sources in the configuration file is called NSRC_SELECT and the values expected are of type integer. Three different options are implemented in CAP for this selection:

- NSRC_SELECT < 0: All possible solutions are considered for the number of sources, from q=1 to M-1, and a frequency wavenumber decomposition is calculated for each of these solutions.
- 1 $<$ NSRC_SELECT $<$ M-1: the number of sources is fixed to NSRC_SELECT. The NSRC_SELECT first eigenvectors of the spectral matrix describe the signal subspace and the M-NSRC_SELECT last eigenvectors describe the noise subspace.
- NSRC_SELECT $= 0$: the number of sources is determined automatically using a statistical approach, based on the theory of information (Akaike 1974).

As already indicated above, there are two distinct processing strategies. MUSIC gives estimates of multiple plane wave characteristics for each individual time-frequency cell, whereas MUSIC2 evaluates for each frequency band the time averaged CSM resulting in a single set of multiple plane wave propagation characteristics. It should be noted, that the window based processing takes a significant amount of computation time.

The output produced by MUSIC resembles closely that of the CVFK implementation. Instead of the semblance value the current number of the multiple maximum is given. The "power" estimate of the MUSIC estimator does not provide a correct power measure, as it is rather a pole location in the slowness map. A direct interpretation of this value should be avoided. The MUSIC2 output is equivalent to the output created for the CAPON or CVFK2 method, the considerations regarding the power estimate hold as stated above. For more details of the output files for MUSIC and MUSIC2 see section 7.2.

5.7 Modified SPatial AutoCorrelation – MSPAC

We have implemented the modified spatial autocorrelation method according to Bettig et al. (2003) in order to use more arbitrary array geometries for the computation of spatial autocorrelation curves.

The MSPAC method is selected by setting the METHOD keyword value to 4. The time frequency selection applies as for the f-k methods, that is time window lengths and frequency bands must be chosen according to section 5.1. The BANDWIDTH parameters is used to construct a narrowband filter in frequency domain by using a cosine taper function around the center frequencies and limits calculated from BANDWIDTH. From the experiences acquired during the tests with both synthetic and real data sets we recommend to use very long windows for the computation of the spatial autocorrelation coefficients. WINFAC should be used instead of WINLEN here and a typical value for WINFAC is 50.

The determination of the ring partitioning from the co-array can be either automatic or manually. The automatic determination of rings, based on the distributional characteristics of radial and azimuthal density within the corresponding co-array, performs reasonably well for sparse co-arrays. Very densely populated co-array geometries usually result in poorly estimated ring partitions as no distinct gaps can be recognized from the distributions. In these cases or for comparison with other software (GEOPSY) it is possible to read the ring partitions from a simple ASCII file formatted as the ".ring" files written by GEOPSY MSPAC processing capability. We recommend to use the GEOPSY interface to determine interactively the ring partitions from the co-array distribution. For the task of repeated computation of autocorrelation curves on one and the same array geometry but for different data portions, simple shells scripts can then by used with **CAP** . For single runs, GEOPSY is preferred, as it contains some advanced functionality (like "antitrigger" pre-processing).

The manual ring selection mode is selected simply by specifying the corresponding ".ring" file with the -r option at the command line. Whenever this option is missing, the automatic ring partitioning is used instead. The output of the averaged autocorrelation coefficients is written to a file with extension ".stmap". Please see section 7.2.3 for details of the format.

The MSPAC processing option within **CAP** contains also the inversion of autocorrelation curves into a dispersion curve. However, this functionality was long time not working as expected and only recently has been reconsidered by A. Köhler. We expect within near future to integrate his work into the current code. There are a number of keywords in the configuration file connected with the non-linear inversion scheme presented by Bettig et al (2003). Those are OMEGA, APRIORI, CR@1HZ, CREXP, BESSMINARG and BESSMAXARG. Meanwhile the full functionality of the inversion scheme is not implemented, we leave these parameters uncommmented here. For those interested in the meaning of these parameters we refer to section 7.1.3 and the appendix A.

Horizontal component processing is implemented now, but is still subject to major revision and should not be used in the moment. Updates are expected in near future though.

5.8 Single station H/V ratio computation

Although the H/V processing issues have been treated in other work packages of the SESAME project and a full functional platform independent Java based processing software has been developed (JSESAME, Atakan et al., 2004) we felt that sometimes it could be convenient for those who are mainly working on the analysis of array data sets to have a look onto the H/V spectral ratios of individual stations within the array setup (e.g. if there are indications for 2D or 3D site effects). For this purpose it would be inconvenient to switch from the main software package to another one. Please note, that this implementation of H/V computation is rather limited regarding the choice of options flexibility of parameter settings or statistical analysis. It includes just the very basic computation of H/V ratios for contiguous time window data (continuous data streams) and provides as output an average H/V curve and variances. For advanced H/V processing, like "antitrigger" window selection, individual window H/V results, statistical tests or improved visualization capabilities please switch to **JSESAME** or to a similar implementation of H/V processing within **GEOPSY**.

The H/V processing option is selected by specifying a value of 7 for the keyword METHOD in the configuration file. One or several stations can be selected for processing, but only stations having all 3 components available in the database are considered for processing (stations containing just a single horizontal components can not be processed). The window length of the indivdual time windows for which the spectral ratios are computed is selected by the keyword WINLEN. The argument to WINLEN is a float value specifying the window length in seconds. Please note, that WINFAC has to be set to a negative number in order to make the parameter for WINLEN to take effect. The progress between time windows in seconds is specified by the value given for the keyword STEP.

For each time window the vertical and two horizontal component wime windows are Fourier transformed using the fftw-3.0.1 software package (http://www.fftw.org). After transformation, amplitude spectra are computed and smoothed additionally using a smoohting window suggested by Konno and Ohmachi (1998). The width of the smoothing window (parameter b in Konno and Ohmachi, 1998) is given by the integer value associated with the keyword KOSMOOTH in the configuration file.

The average of the H/V ratio is computed for each individual station by a recursive sample mean and sample variance estimation procedure described at Wolfram Research[2]. Assuming log-normal distributed spectral amplitudes, sample mean and variance estimation is performed by taking the natural logrithm of the smoothed spectral amplitudes.

Three quantities are computed, the H/V ratio using the geometrical mean of the horizontal components N and E as estimate of H, the ratio N/V and E/V as well as the averaged quantities for Z, N and E spectra. After processing, the averaged values and variances are stored "as is" in an ASCII file. Thus, to obtain the H/V ratios, the values have to be corrected for each discrete

[2]Eric W. Weisstein. "Sample Variance Computation." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/SampleVarianceComputation.html

frequency as:

$$H/V(f_i) = \exp(\overline{H/V}_i \pm \sqrt{\sigma_i^2}) \tag{36}$$

or equivalently

$$\begin{aligned} H/V(f_i) &= \exp(\overline{H/V}_i) \\ H/V(f_i)_{-\sigma} &= \exp(\overline{H/V}_i)/\exp(\sqrt{\sigma_i^2}) \\ H/V(f_i)_{-\sigma} &= \exp(\overline{H/V}_i)\exp(\sqrt{\sigma_i^2}) \end{aligned}$$

For the formatting of the output file, please see section 7.2.

5.9 Supplemental and Experimental Methods

In this section I want briefly give an overview of additionally implemented methods or functionality which has been tested within the context of SESAME. Most of these methods presented here are of experimental nature and are still subject of a more thorough testing and evaluation phase. Due to the experimental nature of these methods, the processing output is not optimized and is probably changed in near future.

5.9.1 Hypothesis testing for pre-selection – HYPTEST

During the course of SESAME we had the feeling, that it might be advantageous to exclude individual time windows from the processing whenever they do not fulfill the assumptions on which the analysis is based. In other words, we try to perform a hypothesis test on the occurrence of Rayleigh wave characteristics within a specific time window in order to restrict the analysis on those time windows passing the test.

Although it is relatively straight forward to qualitatively give criteria for detecting Rayleigh wave propagation characteristics, it has been much harder to implement quantitative tests for the clear presence or absence of Rayleigh waves. Until now we have implemented only two very simple approaches for hypothesis testing. For either one of the tests, the keyword METHOD must be set to the integer value 9. The integer argument of the keyword HYPMETH is then used to determine which hypothesis method shall be selected. Currently allowed values for HYPMETH are 0 or 4 (see below).

The first approach consists in a ridge-detection algorithm similar to a method described by Schissele et al. (2004). It is selected by setting the HYPMETH keword to a value of 4. In this approach the energy content of the signal is evaluated for each frequency band individually for each station and compared with the pointwise estimate of the actual instantaneous frequency. Whenever a waveform sample of a narrowband filtered version of the waveform data exceeds a certain energy threshold level and additionally the instantaneous frequency estimate fomr the broadband waveform at this time fits to the frequency band of interest, the waveform sample is

considered as a potential candidate for further processing. The analysis is taken out station per station and all candidate samples are determined. In a final step a coincidence trigger is applied to the set of array stations and a list of time frequency boxes are allocated for those portions of the waveforms which fullfill all three criteria of the hypothesis test.

The energy threshold criterion which must be met is user selectable and is given a fraction of the maximum energy found in the whole data portion subject to processing. It is specified by setting a value in the range of $[0., 1.]$ for the keyword TFENERGY_TH1. Typically this value is of the order of a few percent (e.g. 0.01 or 0.05). The coincidnce trigger threshold is specified with the keyword TFENERGY_TH2. Again this value can take values in the range of $[0., 1.]$ and translates in the percentage of stations which must show potential waveform candidates at a given time. A value of 0.8 means therefore, that 80% of all stations in the array setting must positively fullfill the energy and instantaneous frequency criteria explained above.

The alternative hypothesis testing approach consists in an "antitrigger" strategy for linearly polarized signals and is selected by giving the value 0 to the keyword HYPMETH in the configuration file. The aim is to remove signal windows containing a strong body wave component from further processing. Thus, a polarization analysis (Jurkevics, 1989) is performed for each individual station and time-frequency cell determined from the settings given in the configuration file (see section 5.1 for details). Whenever the linearity of a waveform portion within a time window exceeds a certain, user selectable, threshold, this window is *excluded* from the list of possible candidate windows for further processing. After the lists of time windos are determined for each station of the array, again a coincidence trigger is used to find the final list of time frequency boxes which are kept for further processing. The tresholds are slected by the keywords TFPOLJURK_TH1 and TFPOLJURK_TH2. The first keyword is used to specify the degree of linearity which must not be exceeded (value range $[0., 1.]$) and the value given to the second keyword again specifies the percentage of stations which must pass the test and is used for the conincidence-trigger test.

For both hypothesis testing methods, the list of time-frequency boxes is written to a file following the formatting of the "tfbox" list files. After this, **CAP** terminates execution. Then, in a second run of **CAP** one can use this file to compute the wavefield propagation characteristics for the pre-determined time-frequency cells. We have chosen to use this approach in order to facilitate the repetition of processing on exactly the same pr-selected data portions with different methods while avoiding the necessity to perform the same hypothesis testing over and over again.

In an earlier stage of the project we had additionally implemented another hypothesis testing approach. However, this works only as an option for the methods CVFK and CVFK_FAST and is restricted to the use of single component data (vertical component for obvious reasons). The idea of this approach was to detect those time windows which contain a single dominant coherent signal contribution as the wave propagation characteristics will then not be biased by the superposition of plane wave portions crossing the array in different directions. The detection of a single dominant signal contribution is achieved by evaluating the eigenspectrum of the cross spectral matrix. The relation between the first and second eigenvalues λ_1/λ_2 is compared to a user selectable threshold specified by the value given for the keyword SINGVAL_RATIO (typical value ≈ 10. Whenever the eigenvalue ratio exceeds this threshold for the cross spectral

matrix evaluated for a specific time frequency cell, the time window is passed directly to the CVFK/CVFK_FAST analysis routines.

This hypothesis testing approach is *NOT* selected by setting the METHOD keyword to a value of 9. In this case, the METHOD must be set to 0 (CVFK) or 14 (CVK_FAST) and the keyword DETECT_DOMINANT has to be set to 1. A value of 0 toggles this pre-processing feature off and all time windows are passed to the analysis routines. A "standalone" version of this hypothesis test exists and can be selected by choosing a value of 8. By doing so, an ASCII output file is created which records the eigenspectra normalized to the largest eigenvlaue (λ_1) for each time window. *Please note that this processing option has just been introduced for testing purposes!*

Our experience with the improvements which can be achieved with these hypothesis methods are rather limited in the moment. Further testing is required and subject of current investigation.

5.9.2 Cross-Correlation Stack – CCSTACK

Campillo and Paul (2003) have shown, that the use of simple cross-correlation stacks on portions of late surface wave coda from regional earthquakes allows to obtain estimates of Green's functions between a pair of stations based on the diffuse scattering theory. Recently, Shapiro and Campillo (2004) reported even, that similar observations are possible for ambient noise.

We have implemented this simple technique to investigate the feasibility for ambient vibration measurements. For all pairwise combinations of a set of channels (station and component selection) time windows are randomly selected from the overall selected time range. Foreach of the time windows the cross correlation is computed and stacked. The number of time windows for stacking is chosen by the keyword NSTACK which expects an integer argument. Setting the COMP keyword to 1, only stacks between the vertical components of the selected stations are computed, whereas using the number 123 results in additional stacks for all combination between Z, R and T components. R and T components are rotated from the horizontal N and E time series into the inter-station direction of each pair. For any other choice of COMP no computation takes place. The length of time windows used for the cross is best given by the use of the keyword WINLEN while setting WINFAC to a negative value (see section 5.1).

One additional option is available for this method. The keyword PREWHITEN can be toggled on or off by using the integers 1 or 0. Using the prewhitening option results in the construction of a FIR filter which tries to whiten the spectra of the time series before computing the cross correlations. The idea behind this procedure is to remove any band limitations of the source signal and the hope was ot obtain a very boradband estimate of the Green's functions. However, also the structural information is apparently suppressed (in the time series the source signal has already propagated through the media), the pure phase information seems not to be sufficient to reconstruct the Greens' functions.o

In the following figure 5.9.2 on page 38 we show a successful example of applying this methods to ambient noise records acquired in the Lower Rhine Embayment. Clearly a surface wave like wavetrain is recognized in the plots. Currently we consider this technique as a promosing

approach to obtain site structures. Further investigation of this technique is needed to prove its feasibility and to get insight into its limitations.

5.9.3 Attenuation estimation – QEST

The attenuation of seismic surface waves plays a quite important role, when it comes to the interpretation of the dispersion characteristics measured from ambient vibration array experiments. Due to the usually high attenuation found in shallow unconsolidated sedimentary structures it is possible, that the energy content of individual modes (especially the fundamental mode) is so strongly damped that we can no longer recognize their contribution in the dispersion properties.

For the problem of attenuation estimation from passive ambient vibration array measurements Zywicki (1999) suggested to obtain attenuation coefficients from fitting a plane through the displacement amplitudes of a 2 dimensional array configuration. We have implemented this approach for testing reasons. The results we have obtained so far are ambiguous. However, we noted that evaluating the distributional information from a large number of individual time window attenuation coefficient estimates (similar to the derivation of dispersion curves from the CVFK algorithm) may lead to a consistent estimate of frequency dependend attenuation properties of the wavefield. We'll report on further experiences with this approach.

In order to use this experimental method, the value for the keyword METHOD has to be set to 11. Time-frequency tiling applies as for the f-k methods as explained in section 5.1.

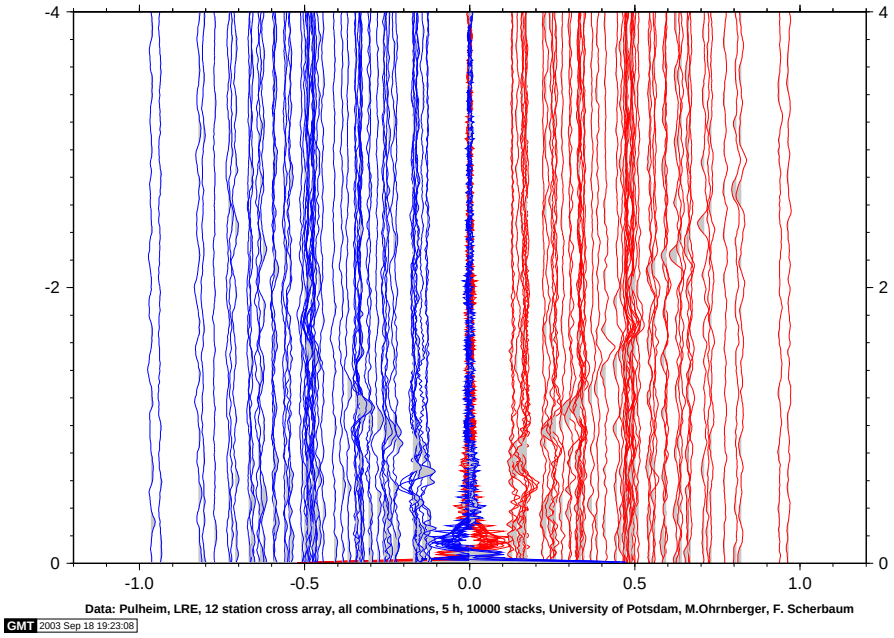

Figure 7: Cross correlation stacks for 5 hour ambient vibration data at site Pulheim in the Lower Rhine Embayment. All vertical components of a 12 element cross array configuration have been used. Forward and time reversed processing have been plotted on positive and negative x-axis.

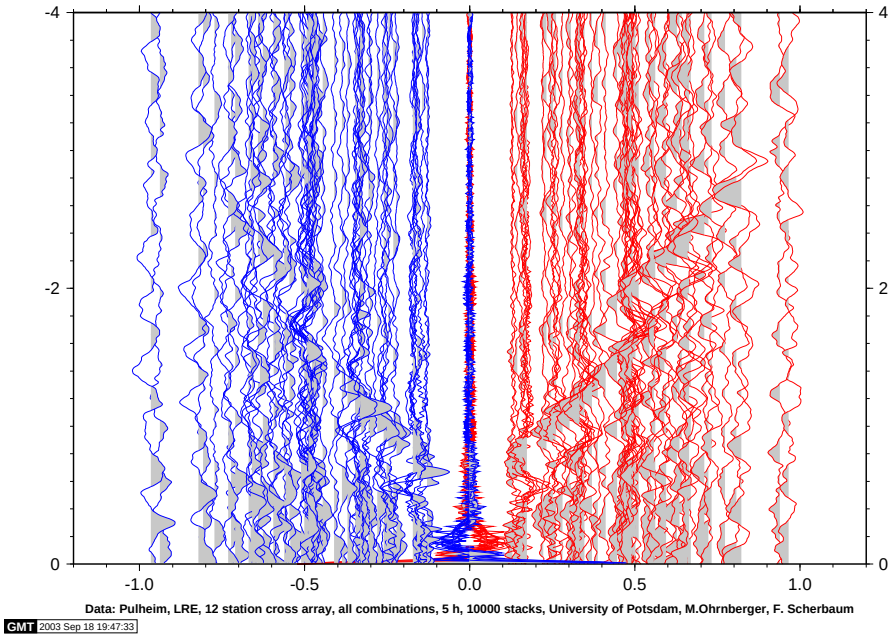

Figure 8: As before, but now wiggle plots are scaled to larger amplitudes with increasing offset.

6 Postprocessing

CAP includes three simple postprocessing options in order to provide measures of confidence for an easier interpretation of the obtained phase velocity results. Two of these options are integrated into the processing flow of **CAP** , one is provided as standalone program utility which acts on output files. These three postprocessing "methods" are:

- slowness response evaluation (SLOWRESP)
- **fk2disp** - standalone tool
- fraction of wavenumber map (MAPFRAC)

The usage of these post processing strategies are discussed in the following

6.1 Slowness response evaluation (SLOWRESP)

The slowness response evaluation can be applied exclusively to the CVFK method. It is based on the comparison of the estimated slowness map with the theoretical array response function computed for the actual frequency band and centered on the obtained wavenumber estimate. The idea behind this is to facilitate the recognition whether a single dominant signal is contained in the analysis window. As the CVFK algorithm shows as is generally known relatively broad maxima in the slowness response function and therefore is said to be relatively bad resolving compared to other f-k estimators. When multiple plane wave arrivals with similar energy content are present in a single analysis window, the CVFK usually fails to separate the individual signal contributions to the wavefield and tends to give biased results. Thus, the recognition of a single dominant signal arrival within one analysis window would allow to put more confidence on the estimates of this time window.

In order to achieve this goal, we compute three quantities. Those are the average semblance values within the complete slowness map, both for the theoretical and observed f-k maps, as well as the squared residual sum taken over all slowness grid points between real and theoretical slowness maps. These quantities may be expressed as:

$$\overline{P}_{real} = \frac{1}{N_{grid}} \sum_{i=0}^{N_{grid}} P_{i,real} \tag{37}$$

$$\overline{P}_{theo} = \frac{1}{N_{grid}} \sum_{i=0}^{N_{grid}} P_{i,theo} \tag{38}$$

$$Res = \frac{1}{N_{grid}} \sum_{i=0}^{N_{grid}} (P_{i,real} - P_{i,theo})^2 \tag{39}$$

We considered the residual sum as well as the ratio between the maximum semblance to the average semblance value (for both observed and theoretical array responses) to derive proxy parameters for the presence of single dominant signal arrivals in a given time window. However, from synthetic tests, the obtained results provided rather ambiguous results. No clear threshold levels could be defined. Variations with frequency as well as with station geometry seem to be too large to allow a generally applicable solution. Although the work on this postprocessing options was discontinued in an early stage of the SESAME project, we still think that it should be possible to use these quantities. In future we think of using a bootstrap analysis to derive acceptable thresholds for a given station geometry and separately for frequency bands.

6.2 Determination of dispersion curves - fk2disp

The main processing utility of **CAP** is a standalone command line tool named "fk2disp". The name tells the purpose of this small program. fk2disp is run on the output files of CVFK, CVFK_FAST and MUSIC processing outputs and determines a dispersion curve from the distribution of wave propagation estimates from the ensemble of time windows within each frequency band. From the distributions, the sample mean, sample variance, median as well as the 25% and 75% quantiles are computed.

As additional feature fk2disp allows for the CVFK / CVFK_FAST processing output files to "filter" the ensemble using simple thresholds on the semblance coefficients and/or beampower values. Please note, that this feature cannot be used for the results obtained from the MUSIC method, as neither semblance values are computed, nor the array estimator provides a real power measure in the grid search. Nevertheless, statistics on the the complete set of wave propagation estimates can be performed.

The usage of this tool is simple. It is called from the command line and takes exactly three arguments. The first argument is the output file of a run of **CAP** using one of the three methods CVFK, CVFK_FAST or MUSIC. Please note, that the header *must not be removed* from the output files before calling fk2disp. The information from the header is used to re-compute grid dimensions, obtain information about the time-frequency settings, etc. The second and third arguments specify thresholds used for the rejection of analysis results from individual time windows showing too low coherency and/or having too small energy to be considered for the dispersion curve estimates. Both threshold are specified as fraction of the maximumal choerency/energy encountered within each individual frequency band separately. A value of 80 means that the semblance/energy must exceed 80% of the maximum value found, that is vlaues larger than 0.8*MAX pass the threshold test.

Calling fk2disp without any argument, displays the syntax:

```
krakatau:/home/mao/cap> fk2disp

Usage: fk2disp <bbfk-max-output> <percentage of max><percentage of powmax>
```

Here an example of running fk2disp:

```
krakatau:/home/mao/cap> fk2disp cap-test.max 20 25
```

Three new ASCII files are created from fk2disp. The extension ".max" is removed from the input file name and new extensions are created automatically. The output file containing the dispersion curve estimate from first order statistics gets the extension ".disp". It contains 14 columns in the folling order:

- 1st column: index of frequency band
- 2nd column: center frequency of band
- 3rd column: minimum semblance coefficient observed for this frequency band
- 4th column: maximum semblance coefficient observed for this frequency band
- 5th column: minimum beampower observed for this frequency band
- 6th column: maximum beampower observed for this frequency band
- 7th column: sample mean
- 8th column: sample standard deviation
- 9th column: sample median
- 10th column: lower quartil of distribution (25% quantil)
- 11th column: upper quartil of distribution (75% quantil)
- 12th column: median deviation
  (median of absolute difference between samples and median)
- 13th column: number of windows exceed given threshold
- 14th column: total number of windows in frequency band

The first line is a header line, as usual indicated by the "#" character at position 1. Here is an example of a ".disp" output file

```
# iFreq freq minthres thres minthres2 thres2 mean std median uqtil oqtil meddev nb  n
0 0.300000 0.084693 0.994377 45.069200 91.933800 0.306683 0.485289 0.222193 0.150407 0.349196 0.101382 211 211
1 0.316285 0.072577 0.993708 49.400800 94.923000 0.292296 0.466573 0.224810 0.146107 0.334408 0.089779 222 222
2 0.333455 0.085690 0.990457 46.216000 90.743700 0.309668 0.433483 0.215381 0.132779 0.343270 0.095476 235 235
3 0.351556 0.084751 0.987854 46.178300 93.293200 0.281429 0.356575 0.212747 0.139487 0.322218 0.085921 247 247
...
...
47 3.598687 0.118651 0.595134 58.755000 83.460500 1.936113 1.042236 1.762435 1.171840 2.700950 0.771607 2546 2546
48 3.794041 0.137799 0.591091 58.048300 82.570400 1.850170 1.002360 1.695290 1.120970 2.549430 0.707630 2685 2685
49 4.000000 0.114876 0.578155 57.140400 80.476100 1.770665 0.990997 1.653585 1.061350 2.393200 0.665779 2830 2830
```

The second output file contains histogram information from the evaluation of the distributional characteristics of all results. For each frequency band a histogram of the slowness/app. velocity estimates is created. The bin-width of the histograms is chosen according to the resolution of the original chosen grid. Three different types of histograms are created and stored in different columns of the output file: 1) the number of observations within each histogram bin; 2) the average semblance computed from the sample mean of all observations falling into one bin; 3) the average beampower computed from the sample mean of all observations falling into one bin.

```
0.300000 0.000000 0.004739 0.880407 61.891300 0 0
0.300000 0.025000 0.014218 0.924183 58.236600 0 1
0.300000 0.050000 0.018957 0.927418 60.544525 0 2
...
0.300000 4.950000 0.000000 0.000000 0.000000 0 198
0.300000 4.975000 0.000000 0.000000 0.000000 0 199
0.300000 5.000000 0.004739 0.087235 76.141900 0 200
0.316285 0.000000 0.004505 0.940595 58.099600 1 0
0.316285 0.025000 0.013514 0.834131 63.489500 1 1
0.316285 0.050000 0.045045 0.893412 61.574750 1 2
...
0.316285 4.950000 0.000000 0.000000 0.000000 1 198
0.316285 4.975000 0.000000 0.000000 0.000000 1 199
0.316285 5.000000 0.004505 0.091419 72.783000 1 200
0.333455 0.000000 0.012766 0.941214 60.913667 2 0
0.333455 0.025000 0.008511 0.722124 59.151700 2 1
0.333455 0.050000 0.046809 0.834642 59.301345 2 2
...
...
4.000000 4.950000 0.000000 0.000000 0.000000 49 198
4.000000 4.975000 0.000000 0.000000 0.000000 49 199
4.000000 5.000000 0.000000 0.000000 0.000000 49 200
```

The last output file has extension ".csh" and is a csh-script which allows to plot the histogram and dispersion information using the GMT software package. Examples of this visualization of f-k analysis results are given in section 7.3.

6.3 Using MAPFRAC for uncertainty bounds

Several of the implemented f-k analysis methods determine a single slowness map for each frequency band. Thus, it is not possible to determine sample mean and variances from repeated estimates as described above for the thos methods which determine one f-k map for each individual time window (CVFK, CVFK_FAST or MUSIC methods). In this case, it is possible to obtain some uncertainty measure by considering not only the single best value, or a fixed number of local maxima of the f-k map as appropriate measure for the wave propagation characteristics.

Using the keyword MAPFRAC it is possible to determine larger regions from the f-k maps, which belong to the "best" estimates of wave propagation. The value given for MAPFRAC gives the fraction of grid points with highest coherence/power values. Setting MAPFRAC to 0.01, for example, keeps 1% of all evaluations and stores those into an extra output file with extension

".best". Note, that no coherence or power threshold is needed. The extracted regions from the wavenumber maps may be contiguous or separated, depending whether multiple maxima are present in the slowness map or not. Figure 6.3 shows an example for a f-k result obtained with Capon's estimator at a frquency of 1.32 Hz. MAPFRAC has been set to 0.05 in this example. The regions enclosed by the thick contour lines show all the grid points which are kept in the ".best" file.

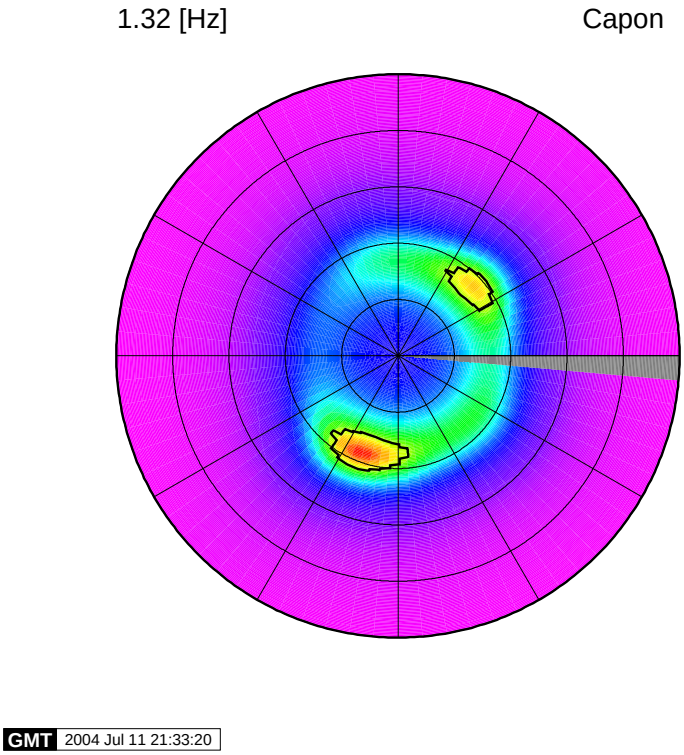

Figure 9: Example for using MAPFRAC to obtain uncertainty estimates from the f-k results of the Capon estimator.

The fraction of grid-points extracted from the slowness maps can be used to obtain uncertainty bounds for each frequency band. Either it is possible to determine sample mean and variance from the ensemble of grid points (with respect to absolute slowness), or one chooses to take the minimum and maximum from the distribution as uncertainty limits. Although this procedure seems visually in agreement to our expertise, we have not yet determined any theoretical foundation for the appropriateness of these uncertainty values. As a consequence, a clear answer to the question which value shall be selected for the use of the MAPFRAC option can not be given so far. Meanwhile, we just can state, the the uncertainty estimates obtained in this way can still provide an adequate weighting scheme for the misfit function of the dispersion curve inversion

problem.

7 Usage

After detailing individual analysis methods and pre- and post-processing options, we want to outline the usage of the software package **CAP** . Therefore, we will first comment on the pre-requisites, that is the input file types which are needed by **CAP** . Subsquently, the different output files and formatting issues of the analysis results are explained in detail before showing some example results obtained with **CAP** . As in the moment there exist three different versions of **CAP** , which differ in the way of accessing the input information, we indicate, where needed, the differences of usage.

7.1 Input files

7.1.1 Supported waveform file formats

Depending on the version of **CAP** , different waveform file formats can be used with **CAP** . The following table summarizes the supported waveform file formats for each version

Version	Supported formats
FAKE_DB	GSE1 and GSE2
GIANT_DB	GSE1, GSE2 and PDAS
GEOPSY_DB	SAC (Little/Big Endian), GSE1, GSE2, SEG-2, SU (Little/Big Endian) Cityshark 1/2, SAF, SISMALP, plain ASCII columns

Other important and widely used file formats in the seismological community (i.e. MSEED) are likely to be readable in a future version of this software.

7.1.2 Waveform list and station file (FAKE_DB only)

For the standalone version of **CAP** , which does neither require an existing GEOPSY nor GIANT database, the information of station geometry, calibration and waveform files is specified by two formatted ASCII files. These files are specified at the command line with the -s option (station/calibration information) and by the -w option (waveform list).

Here is an example of a correctly formatted station file:

```
# CALPATH /scratch/scratch/ARRAY2003/moxa2003/CALIB/
GP01 SHE 0. 0. 0. gp01-e.cal
GP01 SHN 0. 0. 0. gp01-n.cal
GP01 SHZ 0. 0. 0. gp01-z.cal
GP02 SHE 0. 0. 0. gp02-e.cal
GP02 SHN 0. 0. 0. gp02-n.cal
GP02 SHZ 0. 0. 0. gp02-z.cal
GP03 SHE 0. 0. 0. gp03-e.cal
...
```

The station file contains therefore a header line which specifies the path to a directory which contains the calibration files for each station and component in the dataset. It is followed by a single line per station and component pair (considered as one data channel). In each of the lines, the station name is given in the first entry followed by the component information, longitude, latitude and height (in deg, deg, km) and the corresponding calibration file name which is assumed to be formatted in the GSE1 pole and zero notation (PAZ).

An example of a waveform list file is given here:

```
# /scratch/scratch/ARRAY2003/moxa2003/1031105/
081124_0.P04 WID2 2003/11/05 08 11 24.000 GP04 SHZ CM6 228500 125.000000 2.5465e+00 1.000 NONE -1.0 0.0
081124_1.P04 WID2 2003/11/05 08 11 24.000 GP04 SHN CM6 228500 125.000000 2.5465e+00 1.000 NONE 0.0 90.0
081124_2.P04 WID2 2003/11/05 08 11 24.000 GP04 SHE CM6 228500 125.000000 2.5465e+00 1.000 NONE 90.0 90.0
083928_0.P03 WID2 2003/11/05 08 39 28.000 GP03 SHZ CM6 143000 125.000000 2.5465e+00 1.000 NONE -1.0 0.0
083928_1.P03 WID2 2003/11/05 08 39 28.000 GP03 SHN CM6 143000 125.000000 2.5465e+00 1.000 NONE 0.0 90.0
083928_2.P03 WID2 2003/11/05 08 39 28.000 GP03 SHE CM6 143000 125.000000 2.5465e+00 1.000 NONE 90.0 90.0
...
```

The waveform list file contains also a header line (marked with #) which specifies the absolute path to a directory which is prepended to the waveform file name given in the following lines. Each of the following lines containes as first entry the waveform file name (GSE1 or GSE2 formatted). It is followed by the GSE1 or GSE2 header line of the corresponding file.

At startup, **CAP** reads the station, calibration and waveform header information from these ASCII files and stores these information into the internal representation of the database structures. After this initialization, the processing continues as for the GIANT or GEOPSY versions of **CAP** .

7.1.3 Configuration file

The configuration file is a simple ASCII file containing keywords and keyword parameters. The file is kept in a free format allowing for an unlimited number of comments. The only requirement is that the keyword must start at the beginning of the line and is immediately followed by the appropriate parameter.

Three types of keyword functionality can be distinguished: keywords working as switches, keywords setting parameter values, and keywords which provide parameter values and addtionally switch actions depending on the range of parameter value given.

The following table summarizes the available keywords, value types, and allowed value ranges.

Entries in configuration file used with **CAP**				
METHOD	-	0 - 12,14	integer	selects analysis method - main switch
HYPMETH	-	0,4	integer	selects hypothesis testing method for preselection of time-frequency boxes
TFPOLJURK_TH1	0.85	[0., 1.]	float	Threshold for "antitrigger" like polarisation hypothesis testing mode (HYPMETH 0)

Keyword	typical value	value range	Data type	Description
TFPOLJURK_TH2	0.4	[0., 1.]	float	Coincidence threshold for polarization "antitrigger". If the ratio N_{trig}/N_{total} exceeds this value, the time-frequency cell is selected for processing
TFENERGY_TH1	0.9	[0., 1.]	float	Threshold for time-frequency energy pre-selection criteria (HYPMETH 4) for each individual station. If the reltaive energy in current time-frequency cell exceeds this threshold the station is "triggered".
TFENERGY_TH2	0.9	[0., 1.]	float	Coincidence threshold for time-frequency energy pre-selection criteria (HYPMETH 4). If the ratio N_{trig}/N_{total} exceeds this value, the time-frequency cell is selected for processing
PREWHITEN	0	0,1	integer	switch that toggles prewhitening for CVFK and CCSTACK methods
DETECT_DOMINANT	0	0,1	integer	toggles another pre-selection criteria for CVFK processing. Selection criteria is based on the eigenspectra of the corss spectral matrix
SINGVAL_RATIO	10.	$> 1.$	float	Selection threshold for DETECT_DOMINANT. If the ratio of first to second eigenvalue λ_1/λ_2 exceeds the threshold, the time-frequency box is selected for subsequent processing
SLOWRESP	0	0,1	integer	switches computation of slwoness reponse for centered on previously determined maximum in slowness map. 0: toggles off, 1: toggles on. Used for post-processing.
NUM_BANDS	30	> 0	integer	Number of frequency bands to process. Value must be larger than 0.
LOWEST_CFREQ	-	> 0	float	lowest center frequency to process
HIGHEST_CFREQ	-	> 0	float	highest center frequency to process (>= LOWEST_CFREQ
BANDWIDTH	0.1	[0., 1.]	float	fraction of center frequency used as half(!) bandwidth for CVFK, HYPTEST, MSPAC methods: $[f_c - bw, f_c + bw]$. Not used for the methods based on cross spectral matrix CVFK2, CAPON, MUSIC, MUSIC2

Keyword	typical value	value range	Data type	Description
BANDSTEP	-1.	$[-\infty, 0.[$ or $> 1.$	float	factor used to multiply center frequency in order to get the next higher center frequency. If set to negative values, then the frequency spacing between different frequency bands is computed from LOWEST_CFREQ, HIGHEST_CFREQ and NUM_BANDS to equally distributed center frequencies on a logarithmic frequency scale.
SPATIAL_SMOOTH	0	0,1	integer	switches spatial smoothing for methods based on cross spectral matrix computation (CVFK2, CAPON, MUSIC, MUSIC2).
NSRC_SELECT	0	< 0 or 0 or $[1, M-1]$	integer	switches selection method for the determination of number of sources for signal/noise subspace methods (until now: MUSIC). Negative values will compute the full solution. A value of 0 will choose the Akaike criterion for automatically determination of number of sources, values between 1 and M-1 (M number of stations) will computed solution for this number of sources.
OMEGA[3]	-1.	any	float	smoothing for a priori gauss distribution of model parameters for MSPAC dispersion curve inversion - if set less than 0 - a priori information is set to unity matrix
APRIORI[3]	1.	¿ 0.	float	standard deviation of a priori distribution of model parameters for MSPAC dispersion curve inversion - if OMEGA is set less than 0 this parameter is not used.
CR@1HZ[3]	0.6	¿ 0.	float	$c_R(2\pi f)$ at $f = 1Hz$, Rayleigh wave velocity at 1 Hz for initial dispersion curve model (MSPAC) [unit: km/s]
CREXP[3]	0.1	¿= 0.	float	exponent for initial dispersion curve model $c(2 * PI * f) = c(2\pi) * (2\pi f)^{-CREXP}$
BESSMINARG[3]	0.4	any	float	used to determine minimum argument of bessel function for mspac inversion scheme - negative values will impose no limit
BESSMAXARG[3]	3.2	any	float	used to determine maximum argument of bessel function for mspac inversion scheme - negative values will impose no limit

Keyword	typical value	value range	Data type	Description
GRID_LAYOUT	0	0,1,2	integer	selects layout of grid. 0: polar, 1: cartesian, 2: linear
GRID_TYPE	0	0,1	integer	selects grid type. Sampling can be done either in slowness or apparent velocity. 0: slowness, 1: apparent velocity
GRID_MAX	5.	$> 0.$	float	maximum value in grid
GRID_RESOL	201	> 0	integer	number of points for sampling the wavenumber grid. For polar grid layouts, this parameter gives the number of points to be equally spaced between 0 and GRID_MAX. For cartesian grids, this number of points is distributed in x and y directions from -GRID_MAX to GRID_MAX
NPHI	72	> 0	integer	only for polar grid layouts. specifies number of angular steps for grid
LINEAR_PHI	-	$[0., 360.]$	float	gives direction of plane wave propagation. GRID_RESOL number of "grid"-points are equally distributed from -GRID_MAX to GRID_MAX along line pointing to this direction. The angel is given as backazimuth in degrees (N over E).
MAPFRAC	0.05	$[0., 1.]$	float	Fraction of cumulative histogram of sorted values from wavenumber map which is written to .best output file. E.g., a value of 0.05 writes 0.05*grid_size best values from f-k maps.
NSTACK	1000	> 0	integer	number of random stacks to be computed for CCSTACK method
SEED	0	≥ 0	integer	seed for random number generator - 0: select seed from system clock. Any other positive value allows to reproduce random sequences with specified seed.
COMP	1	1,2,3, 22,33, 123	integer	switch for selecting component to process. 1,2,3 are single component vertical, north, east; 22, 33 combine the horizontal components for R and T components projected along wavenumber direction; 123 is for 3 component methods

Keyword	typical value	value range	Data type	Description
WINFAC	$< 0.$ or $> 0.$	10.	float	If set to positive values, WINFAC adjusts window length for processing according to center frequency as WINFAC$\cdot f_c$. For negative values, a fixed window length is chosen for all frequency bands according to WINLEN parameter.
OVERLAP	$< 0.$ or $[0., 1.]$	-1.	float	This parameter controls the overlap of windows selected for processing in the individual frequency bands. If set to negative values, the overlapping is 50% for all frequency bands. Positive values in the range $[0., 1.]$ select constant time shifts for all frequency bands, where 0. results in 50% overlap according to the window length of the highest frequency band (highly oversampled for lower frequencies) and 1. shifts windows always by 50% of the window length of the lowest center frequency to be processed (may cause unprocessed time chunks for higher frequencies)
WINLEN	5.	$> 0.$	float	This parameter is only used if WINFAC is set to negative values. Then WINLEN specifies the window length in seconds used for all frequency bands.
STEP	1.	$> 0.$	float	This parameter is only used if WINFAC is set to negative values. Then STEP specifies the time shift in seconds between consecutive time windows for all frequency bands.
TAPER_FRAC	0.5	$[0., 1.]$	float	Fraction of window length for tapering data windows before Fourier transformation.
KOSMOOTH	30	> 0	integer	Smoothing parameter b as introduced by Konno and Ohmachi (1998) for logarithmic smoothing taper in frequency domain.
STRIKE[4]	-1.	$< 0.$ or $[0., 360.]$	float	Similar to LINEAR_PHI parameter. Applies to SLANTSTACK method, not to CVFK for GRID_LAYOUT set to 2. Negative values determine the angle for projection due to the elongation of the array distribution (suitable for linear arrays).

Keyword	typical value	value range	Data type	Description
OUTPUT_FILE	-	any OS-specific allowable string	string	Specifies basename of output file if not given at command line with -o flag.
OFILE_TYPE	0	0,1	integer	toggles ASCII or binary file writing for output files. binray output not available for all output files.
WRITE_TRACES	0	0,1	integer	switches writing preprocessed waveforms for quality control or check of preprocessing parameters. 0: don't write waveforms, 1: write waveforms
DECIMATE	1	$\leq$ 1 or $[2, ndat/10]$	integer	toggles integer decimation (downsampling by integer factor) on or off. Values ≤ 1 deactivate integer decimation. Values > 2 activate downsampling with specified value.
SEIDL	0	0,1	integer	toggles instrument correction. 0: don't simulate common instrument response for array stations, 1: simulate common instrument reponse using Seidl's (1980) algorithm. Requires existence of GSE1. PAZ calibration files.
FSIM	0.2	$> 0.$ and $<< \mathrm{Nyq}_f$	float	Common corner frequency for instrument simulation in case that SEIDL is set to 1.
HSIM	0.707	$]0., 1.[$	float	critical dampling for simulating a common frequency response in case that SEIDL is set to 1.
BBP_FILTER	0	0,1	integer	toggles usage of Butterworth bandpass filter for pre-filtering all data before processing (really useful?). 0: don't filter, 1: filter
BBP_LOW	0.2	$> 0.$	float	lower corner frequency for Butterworth bandpass filter.
BBP_HIGH	5.	$> 0.$	float	upper corner frequency for Butterworth bandpass filter.
BBP_ORDER	2	$[1, 9]$	integer	Order of Butterworth bandpass filter. The integer value specifies the number of sections (conjugate complex pole pair - 6dB roll-off) for filtering.
ZERO_PHASE	1	0,1	integer	toggles zero-phase filtering on or off. Zero phase filtering is achieved by forward-backward filtering the time series, thus the number of section of the effective filter is doubled (12dB per conjugate complex pole pair).

Keyword	typical value	value range	Data type	Description
GAUSSNOISE	-1.	$< 0.$ or $]0., x]$	float	If set to negative values, this parameter is ignored. For positive values, gaussian noise is added to the waveforms. The variance of the random process is set to the factor given by GAUSSNOISE mulitplied by the variance of the waveform trace determined from the complete time window.
TIME_CORR	0	0,1	integer	switches the use of timing corrections for individual stations. 0: don't correct times, 1: correct times. Program will ask interactively for time corrections for individual stations. Must specify stationlist as NAM1+NAM2+NAM7+... and shift times.
3DCORRECT	0	0,1	integer	switches correction for array setups on sttep slopes (volcanic environments). 0: don't correct, 1: correct. Correction is achieved by fitting the best inclined plane through the station configuration. Resulting slowness grid (better: shift times) is (are) then computed for this inclined plane. Results have to be interpreted in this coordinate system (South is then downslope!).

Table 1: Overview of configuration file parameters for the use with **CAP**

7.2 Output files

Output files in **CAP** are kept as simple as possible. Since the beginning of the development several changes have been made to one or the other output files. Most changes were necessary to simplify the output and to extend the information given and most of those have been made on request by user's of the software. It is therefore most certain that changes will also take place in the future, although certainly there will be no change without need.

CAP creates several output files with fixed extensions added to a given basename, either provided as argument from the command line switch "-o" or (if this switch is not used) by the argument of the keyword OFILE_NAME in the configuration file.

Output files can be ASCII or Binary format (selected by keyword OFILE_TYPE). As storage space has become inexpensive these days I recommend the use of the ASCII file format due to platform-independence readability and easy check of results. No supplemental programs for binary file conversion into ascii files is supplied with this software package.

Depending on the selected analysis method, files can contain different information, although they have the same extensions. Please note, that we have restricted this section to the documentation of the main

output files. Outputs created from experimental or supplemental methods are not yet described, as we expect the output formats to be re-arranged and cleaned from debugging or other overhead information.

7.2.1 The .tfbox file - keeping track of analysed data

After querying the database and reading all configuration parameters, **CAP** computes in a first step the start and length of individual time windows as well as the lower and upper frequency limits for each indivdual frequency band and stores these values in a list. In order to keep track of time windows processed and to allow reprocessing of exactly these time windows, this information is written into a simple ASCII file. The format of this ASCII file consists of a header line starting with a '#' symbol, where the total number of time windows is specified as well as the number of frequency bands used.

For each analysis window, a single line containing four entries is written. The columns contain start time relative to the absolute start time given, the length of the time window in seconds, the lower frequency limit and the upper frequency limit. The order of entries is such, that all analysis windows for one frequency band are written consecutively. Here is a sample tfbox-output file:

```
# nitem: 11016 nbands: 50
0.000000 10.000000 0.450000 0.550000
5.000000 10.000000 0.450000 0.550000
10.000000 10.000000 0.450000 0.550000
15.000000 10.000000 0.450000 0.550000
20.000000 10.000000 0.450000 0.550000
...
300.000000 10.000000 0.450000 0.550000
305.000000 10.000000 0.450000 0.550000
310.000000 10.000000 0.450000 0.550000
315.000000 10.000000 0.450000 0.550000
0.000000 9.584503 0.469508 0.573843
4.792251 9.584503 0.469508 0.573843
9.584503 9.584503 0.469508 0.573843
...
306.704094 9.584503 0.469508 0.573843
311.496345 9.584503 0.469508 0.573843
316.288597 9.584503 0.469508 0.573843
0.000000 9.186270 0.489862 0.598720
4.593135 9.186270 0.489862 0.598720
9.186270 9.186270 0.489862 0.598720
13.779404 9.186270 0.489862 0.598720
...
298.553759 9.186270 0.489862 0.598720
303.146894 9.186270 0.489862 0.598720
307.740029 9.186270 0.489862 0.598720
312.333164 9.186270 0.489862 0.598720
316.926298 9.186270 0.489862 0.598720
0.000000 8.804583 0.511097 0.624675
4.402291 8.804583 0.511097 0.624675
8.804583 8.804583 0.511097 0.624675
13.206874 8.804583 0.511097 0.624675
...
```

7.2.2 The .max file - main output file

The main output file(s) of **CAP** contain header information followed by the analysis results in plain ASCII. All header lines are indicated by a "#" mark. Depending on the chosen analysis methods the "results"-section differ one from another reflecting the different processing procedures.

We give first an example of the header lines which are common for all methods. The first lines of the header contain information of the time of the program start and, in case the program was started from the shell prompt, the command line supplied at program start.

```
# Start of processing at Tue Feb 24 22:37:29 2004
# cap was started with the following comannd line
# /ldat1/mao/CLUSTERSOFT/sesarray_1.5.8/bin/cap_giant -i DES4000_cvfk.cfg
-g 4001+4002+4004+4005+4006+4007+4008+4009 -f 20000410010030.000 -l 20000410015930.000 -o DES4000_cvfk
# We got the following parameters from the configuration file
# METHOD 0
# HYPMETH 0
...
```

In order to allow exact repetitions of program runs, these lines are followed by reporting all parameters read from the specified configuration file.

```
...
# We got the following parameters from the configuration file
# METHOD 0
# HYPMETH 0
# TFPOLJURK_TH1 0.900000
# TFPOLJURK_TH2 0.100000
# TFENERGY_TH1 0
# TFENERGY_TH2 0
...
# BBP_LOW 0
# BBP_HIGH 5
# BBP_ORDER 2.000000
# ZERO_PHASE 0
# List of frequency bands selected for processing:
# Number of freq bands: 40
# Band 0 lower 0.180000 center 0.200000 upper 0.220000
....
```

The next information provided regards the selection of frequency bands as determined from the parameters given in the configuration file.

```
...
# List of frequency bands selected for processing:
# Number of freq bands: 40
# Band 0 lower 0.180000 center 0.200000 upper 0.220000
# Band 1 lower 0.192943 center 0.214381 upper 0.235819
# Band 2 lower 0.206816 center 0.229796 upper 0.252776
# Band 3 lower 0.221687 center 0.246319 upper 0.270951
# Band 4 lower 0.237628 center 0.264031 upper 0.290434
```

```
# Band 5 lower 0.254714 center 0.283016 upper 0.311318
...
# Band 36 lower 2.192276 center 2.435862 upper 2.679448
# Band 37 lower 2.349911 center 2.611012 upper 2.872113
# Band 38 lower 2.518881 center 2.798756 upper 3.078632
# Band 39 lower 2.700000 center 3.000000 upper 3.300000
# Now here is the result...
...
```

Finally, parameters deduced from the configuration file settings are written (e.g. BANDSTEP in the example above) followed by the list of stations which have been selected for processing and the corresponding start times as well as the common sampling rate of the data streams.

```
...
# Now here is the result...
# Bandstep: 1.071905
# List of stations selected for processing:
# start stat 4001 chan 1 20000410010030.005
# start stat 4002 chan 1 20000410010030.005
# start stat 4004 chan 1 20000410010030.005
# start stat 4005 chan 1 20000410010030.005
# start stat 4006 chan 1 20000410010030.005
# start stat 4007 chan 1 20000410010030.005
# start stat 4008 chan 1 20000410010030.005
# start stat 4009 chan 1 20000410010030.005
# Common sampling frequency: 200.000004
# wstart julsec1970 | cfreq | slow/appvel | baz | math-phi | semblance | beampow
955328430.000000 0.200000 0.275 -25 115 0.988025 -200.797
...
```

The "results" section of the output file depends on the the selected analysis approach. We give in the following examples for the output of CVFK (CVFK_FAST) methods, the MUSIC method, whose output resembles closely the one written by CVFK and finally the common output format for the CVFK2, CAPON and MUSIC2 methods.

The CVFK and CVFK_FAST method results are stored in the following way. For each analysed time frequency box a single line is written. The first column gives the start time of the analysis window in Julian seconds (seconds since 1.1.1970). It is followed by the center frequency of the actual frequency band processed. The 3rd and 4th columns contain the estimated plane wave propagation parameters for the most coherent signal in the time window, that is the slowness in units of s/km (or apparent velocity in km/s depending on the chosen GRID_TYPE, see section 5.2) and the direction of signal arrival (backazimuth, points to signal source and is given in degrees measured from north direction via east). For convenience the direction of wave propagation is also given as mathematical angle in degrees (counterclockwise, east equals 0) in the 5th column. The 6th and 7th column finally record the observed semblance value and the beampower of the most coherent signal arrival within the slowness map.

```
...
# wstart julsec1970 | cfreq | slow/appvel | baz | math-phi | semblance | beampow
955328430.000000 0.200000 0.275 -25 115 0.988025 -200.797
955328455.000000 0.200000 0.225 -15 105 0.991457 -198.641
955328480.000000 0.200000 0.5125 90 0 0.982157 -209.03
```

```
955328505.000000 0.200000 0.375 15 75 0.987047 -209.541
...
955329922.669975 0.214381 0.4375 -45 135 0.988642 -203.478
955329945.992944 0.214381 0.5125 -35 125 0.986508 -201.284
955329969.315912 0.214381 1.225 -75 165 0.945696 -210.542
...
955330975.736785 0.229796 0.45 -20 110 0.987285 -201.8
955330997.495219 0.229796 0.4 -55 145 0.988843 -203.42
955331019.253653 0.229796 0.45 -5 95 0.98749 -204.561
...
955331657.517273 0.246319 0.2625 -10 100 0.979455 -197.425
955331677.816124 0.246319 0.475 -30 120 0.985027 -199.236
955331698.114975 0.246319 0.2125 -15 105 0.984514 -200.601
...
...
955331958.333249 3.000000 1.175 -160 250 0.791293 -201.908
955331959.999916 3.000000 1.1625 -160 250 0.763932 -201.527
955331961.666582 3.000000 1.0375 -180 270 0.522684 -204.605
955331963.333249 3.000000 1.1625 -145 235 0.740395 -200.53
# End of processing at Wed Feb 25 00:20:58 2004
```

Choosing the option SLOWRESP for the CVFK method will result in 3 additional columns in the output file. Those contain the average semblance of the observed and the theoretical slowness maps (8th and 9th columns) and the squared residual sum as defined in section 6 in the last column.

The output file written for the MUSIC analysis method resembles closely the output of the CVFK method. The only difference consists in the last two columns. As MUSIC allows to determine multiple plane wave arrivals, the column containing the semblance coefficient in the CVFK (6th column) is used to specify which maximum is actually recorded in the output. The last column in the MUSIC case is just a dummy value and is set to 0 always. It is kept for compatibility reasons to allow a statistical evaluation of this output file by means of the standalone utility fk2disp.

The first five columns of the result sections for CVFK2, CAPON and MUSIC2 methods are equivalent to the entries in the CVFK, CVFK_FAST and MUSIC output files. However, in this case, the results are obtained for each frequency band, not for indivdual time windows within frequency bands. For the CVFK2 and CAPON methods, 4 lines are written for each frequency band. The last column in the CAPON and CVFK2 outputs specify the power estimated for the local maximum in the slowness map. The first line contains the first maximum extracted from the slowness map. It is followed by 3 lines commented by the "#" symbol, containing the three largest local maxima obtained from the slowness map *after* smoothing. In case that less than 3 local maxima can be determined, the power values are set to -1 indicating that these are no valid estimates from the slowness maps. Here is an example:

For the MUSIC2 output, one line for each source within each frequency is written. The amount of lines per frequency band depends therefore on the number of sources estimated (either automatically determined or fixed by the user). An example of the output is given here:

```
...
# start stat S9 chan 1 20000906140000.000
# Common sampling frequency: 124.999994
968248800.000000 0.300000 0.450000 -100.000000 190.000000 0
968248800.000000 0.314434 0.250000 -5.000000 95.000000 0
968248800.000000 0.329562 0.375000 -40.000000 130.000000 0
968248800.000000 0.345419 0.450000 45.000000 45.000000 0
```

```
968248800.000000 0.345419 4.975000 -110.000000 200.000000 1
968248800.000000 0.362038 0.300000 -35.000000 125.000000 0
968248800.000000 0.362038 3.675000 145.000000 305.000000 1
968248800.000000 0.379457 0.175000 -35.000000 125.000000 0
968248800.000000 0.379457 4.975000 70.000000 20.000000 1
968248800.000000 0.397713 0.075000 -75.000000 165.000000 0
968248800.000000 0.397713 4.975000 70.000000 20.000000 1
968248800.000000 0.416849 0.350000 25.000000 65.000000 0
968248800.000000 0.416849 4.975000 -100.000000 190.000000 1
968248800.000000 0.436905 0.200000 5.000000 85.000000 0
968248800.000000 0.436905 4.175000 20.000000 70.000000 1
...
...
968248800.000000 2.862286 0.300000 -45.000000 135.000000 1
968248800.000000 2.862286 0.550000 105.000000 345.000000 2
968248800.000000 2.862286 1.850000 -15.000000 105.000000 3
968248800.000000 3.000000 0.275000 -55.000000 145.000000 0
# End of processing at Sun Apr 18 15:26:03 2004
```

In this case the last column specifies the actual index of the source (numbering starts at 0). The source numbers are sorted according to their corresponding peak height in the slowness maps.

For the H/V processing option the main output file contains the average spectral ratios, variances as well as average amplitude spectra and corresponding variances computed for each individual station from the station list. In the first line, a comment (first character "#") is given about the number of processed time windows. Subsequently the analysis results are output, one line for each discrete frequency consisting of 12 columns. The order of columns is as follows:

- 1st column: sample mean of $\log(\sqrt{(}NE)/Z)$,
- 2nd column: sample variance of $\log(\sqrt{(}NE)/Z)$,
- 3rd column: sample mean of $\log(N/Z)$,
- 4th column: sample variance of $\log(N/Z)$,
- 5th column: sample mean of $\log(E/Z)$,
- 6th column: sample variance of $\log(E/Z)$,
- 7th column: sample mean of $\log(Z)$,
- 8th column: sample variance of $\log(Z)$,
- 9th column: sample mean of $\log(N)$,
- 10th column: sample variance of $\log(N)$,
- 11th column: sample mean of $\log(E)$,
- 12th column: sample variance of $\log(E)$,

A sample of the results section of an H/V ".max" output file is given here:

```
...
# start stat S005 chan 3 20000906140000.000
# Common sampling frequency: 124.999994
# STACKED 15 windows for Station S000
0.991020 2.021358 10.114647 547.371931 0.676377 0.000503 11.545942 1.413773 12.851605 1.397312 12.222319 0.314733
1.028793 1.255309 5.660517 95.941033 0.736154 0.030793 11.671434 0.409452 12.992866 2.266707 12.407587 0.802208
1.584295 1.852071 14.509489 1106.164001 1.300074 0.077696 11.432843 0.297416 13.301358 3.198107 12.732917 1.036916
1.864077 1.525269 12.589566 198.047993 1.573252 0.076972 11.595600 0.357330 13.750503 2.819797 13.168853 0.906569
1.513729 1.190796 8.098397 117.118326 1.150388 0.067599 12.321846 0.396729 14.198917 1.665132 13.472234 0.735760
1.059200 0.554916 3.621308 5.360906 0.711242 0.022630 12.939027 0.226766 14.346186 1.242066 13.650269 0.435508
...
```

The main output of MSPAC processing is not written to the ".max" file, but rather to the output file with extension ".stmap".

7.2.3 The .stmap file - slowness maps

The slowness maps obtained from the f-k analysis methods are written in a simple ASCII output file but unfortunately the file format varies depending on the employed method when running **CAP** .

The table given below specifies the contents and formatting for the individual methods followed by an example of a ".stmap" file for a CVFK run of **CAP** .

Method	number of columns	contents
CVFK	3	frequency, sample mean of semblance coefficient for all time windows, sample variance of semblance for gridpoint
CVFK2	4	frequency, grid index 1, grid index 2, beampower for gridpoint
CVFK_FAST	-	empty file - no grid based computation
CAPON	4	frequency, grid index 1, grid index 2, power estimate for gridpoint
MUSIC	3	frequency, sample mean of "power" estimate for all time windows, sample variance of "power" estimate
MUSIC2	3	frequency, "power" estimate for gridpoint, dummy value -1
MSPAC	7	ring index, frequency index, center frequency of band, average autocorrelation coefficient, variance of autocorrelation coefficient, minimum radius of current ring, maximum radius of current ring

```
0.3 0.330869 0.221395
...
0.3 0.330533 0.221281
...
0.316285 0.24768 0.186334
....
0.316285 0.221705 0.172552
```

```
...
...
1.18584 0.145294 0.124184
1.18584 0.146577 0.125092
...
...
4 0.136087 0.117568
4 0.132529 0.114965
```

It should be noted that for the CVFK and MUSIC/MUSIC2 outputs, there is no information supplied about the position of the semblance/power values in the f-k map. The reason for doing so was to avoid an unnecessary large file size. The position can be decoded as the lines are written always in the same order, that is the second grid index runs in the inner loop while exporting to the ASCII file and the first grid index is used for the outer loop. Thus, for radial grid layouts, you will expect to have for each radial step (slowness) all azimuthal steps written successively. Grid dimensions and reoslution have to be constructed from the header lines of the corresponding ".max" file. Indications how to reconstruct the slowness maps from the ".stmap" file can be found in the shells scripts provided for plotting the slowness maps.

Unexpectedly you will find the main output of the MSPAC computations also in the files with extension stmap instead of the ".max" output files. This unfortunate mixing of outputs will be removed in near future. Meanwhile, however, keep in mind that ".stmap" files contain the autocorrelation curves obtained from MSPAC processing and the extension is expected when using the na_viewer utility from the inversion package sesarray (Author: Marc Wathelet). An example of the output of MSPAC computation is given below:

```
0 0 0.900000 0.960143 0.003903 0.032710 0.042770
0 1 0.910476 0.958683 0.005126 0.032710 0.042770
0 2 0.921074 0.955217 0.007739 0.032710 0.042770
...
...
0 67 1.954241 0.574066 0.103477 0.032710 0.042770
0 68 1.976988 0.552718 0.107464 0.032710 0.042770
0 69 2.000000 0.538375 0.108469 0.032710 0.042770
1 0 0.900000 0.935608 0.006713 0.046180 0.057120
1 1 0.910476 0.933875 0.008033 0.046180 0.057120
1 2 0.921074 0.930695 0.009649 0.046180 0.057120
...
...
1 67 1.954241 0.308452 0.118524 0.046180 0.057120
1 68 1.976988 0.284636 0.116764 0.046180 0.057120
1 69 2.000000 0.255696 0.113950 0.046180 0.057120
2 0 0.900000 0.900336 0.023405 0.060890 0.073020
2 1 0.910476 0.898947 0.021818 0.060890 0.073020
2 2 0.921074 0.895688 0.019151 0.060890 0.073020
...
...
5 67 1.954241 -0.387728 0.146324 0.123110 0.152640
5 68 1.976988 -0.389203 0.139030 0.123110 0.152640
5 69 2.000000 -0.393235 0.131895 0.123110 0.152640
```

7.2.4 The .best file - enable statistics

The output file with extension ".best" stores the output of the N highest f-k map values obtained for each frequency band in the folowing format:

- 1st column: center frequency of processed frequency band.
- 2nd column: slowness (app. velocity) in s/km (km/s).
- 3rd column: backazimuth in degrees (N = 0, E = 90).
- 4th column: signal arrival direction in degrees - mathematical angle convention (counterclockwise, E = 0, N = 90).
- 5th column: dependent on method either semblance value (e.g. CVFK) or power estimate (e.g. CAPON).

A sample of a ".best" file is given below:

```
2.000000 0.75 130 320 0.0434751
2.000000 0.2 135 315 0.0434854
2.000000 0.25 145 305 0.043514
...
...
7.598715 0.675 120 330 0.107147
7.598715 6.1 15 75 0.107249
7.598715 2.875 30 60 0.107293
...
20.000000 6.625 -120 210 0.037592
20.000000 4.125 -155 245 0.0395821
20.000000 4.15 -155 245 0.042568
```

The number N of samples stored form computed slowness maps is determined as fraction of the total number of grid points in the f-k map given by the keyword MAPFRAC in the configuration file (see also section 6). Please note, that for those methods, which determine a f-k map for each individual time window (CVFK and MUSIC), the storage amount can grow enormously for long time series. Consider to set MAPFRAC to very low values (or even 0.) in order to prevent problems with disk space.

7.2.5 The .csh file - plotting your results

CAP writes executable shell scripts (tcsh) to allow a quick visualization of processing results. This is achieved by extracting the information of the main output files by "gawk" and plotting postscript files with the software package GMT by Wessel and Smith (1998).

The output files created are method specific, as the output files differ in format according to the processing method and options. Please note, that the plotting results are not intended for publishing purposes. There are way too many options, the results could be displayed. However, by studying the shell scripts it should be straightforward to modify the plots to resemble the user's needs and preferences. Plotting results are shown in the examples section 7.3

Please note, that for execution of the shell scripts, a fully functional tcsh environment must be available (windows users should consider cygwin) and the GMT software package must be installed. Paths and environemnt variables for GMT must be set correctly

7.2.6 Outputs on stderr and stdout

In order to be track malfunction during data processing, **CAP** writes error messages, warnings and parts of processing information on the stderr / stdout channels of the system. For the GEOPSY version of **CAP** , these outputs must be re-directed into a file when using the command-line interface of **CAP** . Using the GUI interface this output is suppressed.

7.3 Examples

In the following we want to provide some examples of processing options and results obtained from ambient vibration analysis.

7.3.1 Command line usage with GIANT

the GIANT version of **CAP** is called cap_giant and is a pure command line driven program. Calling cap_giant without any option causes the program to output the list of command line parameters and the syntax of usage:

```
============================================================
USAGE: cap -optflag optarg -optflag optarg ...

optionflags:
        -f: <arg: Last time string, 1997[07[20[01[02[21[.32]]]]]]>
        -l: <arg: Last time string, 1997[07[20[01[03[21[.32]]]]]]>
        -i: <arg: parameter settings file name / see below>
        -g: <arg: station list, separated by '+', e.g. B01+B02+B03+B04>
       [-o: <arg: basename for output files>]
       [-t: <arg: file containing time-frequency boxes for processing>]
       [-a: <arg: file containing MSPAC AC-results> -- starts inversion]
       [-r: <arg: file containing ring definitions for MSPAC processing]
        -h: <noarg: print this Help message>


 Environment variable CAPHOME not set -
 set CAPHOME to home of your cap installation
 and run again - then you'll get a printout of
 a cap sample configuration file (sample.cfg)!

============================================================
```

The options given in brackets are optional, the ones without brackets are needed for a successful run of **CAP** . Before starting the program at command line the user has to specify the target GIANT database name, the location of database files as well as path variables for the waveform data and calibration information. GIANT uses internally only relative path and file names within the file system tree. Thus it is possible to move waveform files, database files or calibration information freely within the file system. The specification of path variables is achieved through the use of environment variables, a typical procedure for many programs. The following environment variables have to be set:

IATSN_BASE	name of the database
DBDPATH	points to the location of the database files. The absolute or relative path given here + "/" + the database name and extension ".dbd" must not exceed 39 characters.
DBFPATH	points to the same location as DBDPATH. The same restrictions apply for the length of the overall path names as for DBDPATH
IATSN_DATA	points to the directory from which the data can be accessed by appending the relative path names stored within the GIANT database
IATSN_CAL	points to the directory from which the calibration file information can be accessed by appedning the relative path and files names stored within the GIANT database
IATSN_TMP	points to a directory where temporary data might be written - in case this variable is not specified, the current working directory is assumed. For **CAP** this directory is only used for writing pre-processed waveforms for control purposes.

Depending on the user SHELL environment variable (e.g. tcsh or bash / csh or sh) the syntax for setting environment variables differs slightly. For (t)csh users:

```
krakatau:/home/mao> setenv IATSN_BASE mybase
krakatau:/home/mao> setenv DBDPATH /my/data/base/location
krakatau:/home/mao> setenv DBFPATH /my/data/base/location
krakatau:/home/mao> setenv IATSN_DATA /my/data/waveforms/
krakatau:/home/mao> setenv IATSN_CAL /my/data/calib/
krakatau:/home/mao> setenv IATSN_TMP /tmp
```

and for (ba)sh users:

```
mao@krakatau:~> export IATSN_BASE=mybase
mao@krakatau:~> export DBDPATH=/my/data/base/location
mao@krakatau:~> export DBFPATH=/my/data/base/location
mao@krakatau:~> export IATSN_DATA=/my/data/waveforms/
mao@krakatau:~> export IATSN_CAL=/my/data/calib/
mao@krakatau:~> export IATSN_TMP=/tmp
```

We have assumed throughout that the GIANT database has been created and exists. For details of the creation of a GIANT database, qw refer the reader to the Giant manual.

After specifying the database settings, the user decides which method to apply and which parameters to choose by editing some configuration file (arbitrary file name) to his/her own needs. The user must supply the configuration file, define the start and end times of the data portion which he/she wants to process and further decide on a list of stations, which contain data in this time window. Further it is recommended to use a descriptive name for the output file name. Only the basic name is supplied, extensions are appended automatically by **CAP** . Here is an example:

```
mao@krakatau:~> cap_giant -i myconfig.cfg -f 20020514162300.000 -l 20020514164539.12
-g P001+P002+P003+P005+P010 -o mymethod-mydata-window1
```

During processing, several comments are written to stderr, e.g. regarding the success or failure to access the waveform data through the database, and upon successful completion of the program run, output

files with extensions ".max", ".tfbox", ".stmap", ".best" and "_cap.csh" are created (see section 7.2 for details).

In the following we show some examples of analysis results obtained for a dataset acquired in the Lower Rhine Embayment in NW-Germany. The array configuration consisted in a 12 element cross array configuration with an aperture of approximate 1km. One hour continuous data was processed. All plots (except the MSPAC results) have been visualized by using the automatically generated csh-scripts.

The first two examples (figures 10 and 11) have been additionally edited to contain the theoretical dispersion curves (fundamental and 1st higher mode) derived from a general velocity model for this region (compare Scherbaum et al., 2003).

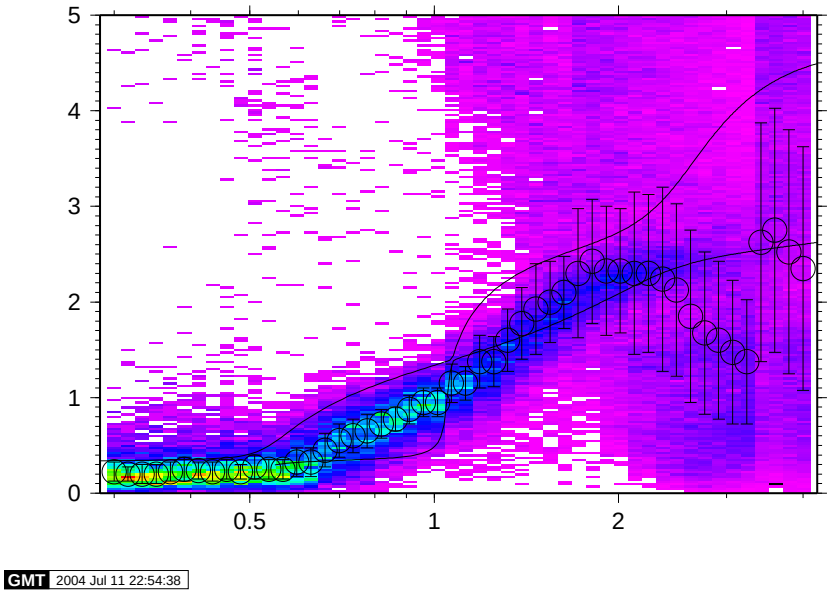

Figure 10: Results of CVFK processing of an array data set acquired in the Lower Rhine Embayment (site Pulheim).

Both figures show the histogram distributions derived from the ensemble of results obtained for each individual time-frequency cells. The histograms were computed using the utility tool fk2disp with both thresholds set to 0. The median is displayed as an open circle and error bars correspond to the median deviation. Symbol sizes scale generally with the percentage of windows exceeding both thresholds, therefore, in this example we have a constant symbol size (all windows have been kept) for all frequency bands.

For the analysis we have used (as for all other plots shown here) a 50% window overlap in all frequency bands and the window length has been set variable containing 10 periods of the center period for each frequency band processed (WINFAC 10 and OVERLAP -1). 50 frequency bands have been computed from 0.3 Hz to 4.0 Hz with a bandwidth of 0.1 (NBANDS 50, LOWEST_CFREQ 0.3, HIGHEST_CFREQ 4.0, BANDWIDTH 0.1, BANDSTEP -1).

For the CVFK (gridbased) method, the grid size was chosen to be porportional to slowness. The polar layout of the f-k grid was computed until a maximal slowness of 5 s/km, 201 points were used for the radial resolution and 72 angle steps were used (GRID_TYPE 0, GRID_LAYOUT 0, GRID_MAX 5,

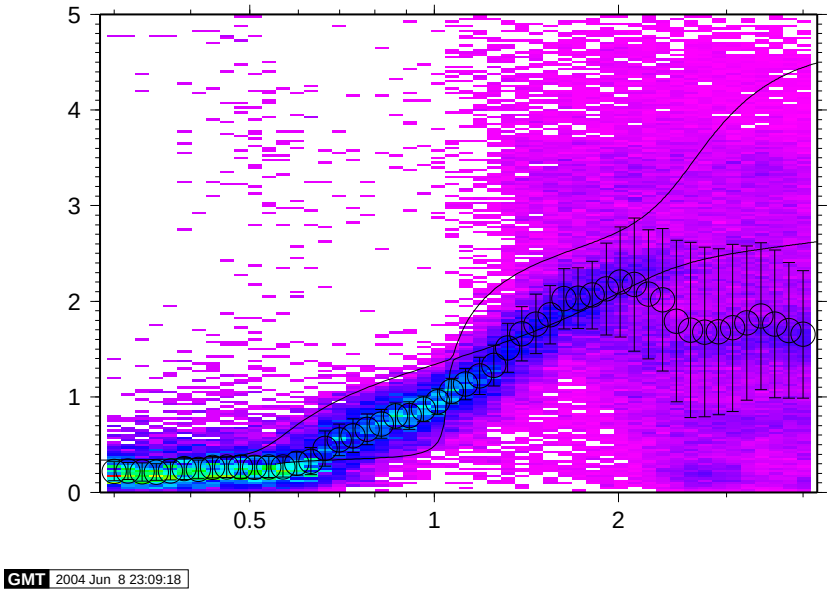

Figure 11: Results of CVFK_FAST processing of an array data set acquired in the Lower Rhine Embayment (site Pulheim).

GRID_RESOL 201, NPHI 72). For the CVFK_FAST, the same parameters were chosen in the configuration file, but only the GRID_MAX takes effect in this case. However, the sampling of the histograms follow nevertheless the slowness resolution given in the configuration file.

Please note, that both results are in very good agreement one with another. The histograms show a clear maximum for frequencies up to 1.5 - 1.8 Hz. The median follows as expected the ridge of the distributions until aliasing occurs. Then the median estimates tend to larger slowness values when compared to the maximum of the distributions.

Comparing the results of the CVFK2 output (figure 12 we can see, how the regions defined from the MAPFRAC option (set to 0.05, compare section 6.3) define some confidence regions for the obtained dispersion curve. The way of displaying CVFK2 results is analog to the following two figures for the analysis results via the Capon estimator (figure 13) and method MUSIC2 (figure 14).

The results of methods CVFK2, CAPON and MUSIC appear to be very similar in this example. In general we have observed that CVFK2 has larger uncertinaties, whereas MUSIC2 sometimes gives unstable results when compared with the more robust CAPON and CVFK2 analysis procedures.

The last f-k method is the MUSIC algorithm (figure 15). The plot given here has been obtained for a much smaller portion of the data window (5 minutes instead of 1 hour) as the processing time requirement is much larger. From the experiences made so far, we find the MUSIC method least suitable for ambient noise analysis. It is computationally expensive and therefore leads to large computation times and the determination of the number of sources, which is a requirement to obtain reliable and highly resolved wavenumber estimates, seems to be relatively unreliable. We specualte that this is due to the strong deviation of the ambient wavefield properties from the assumptions made for the use of AIC of MDL algorithms (i.e. noise subspace does not consist of gaussian noise, Wax and Kailath, 1985).

Finally, we give an example of the processing result from the MSPAC method for the same data set. 6

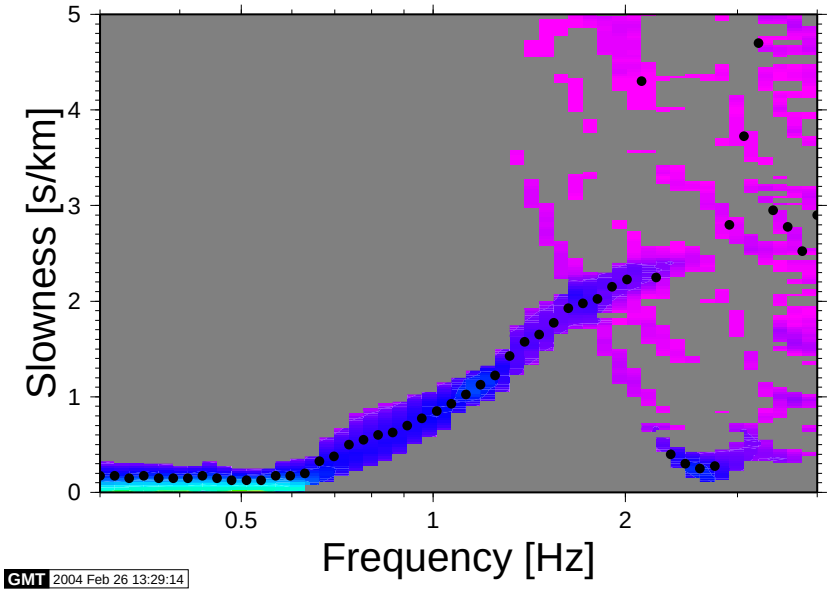

Figure 12: Results of CVFK2 processing of an array data set acquired in the Lower Rhine Embayment (site Pulheim).

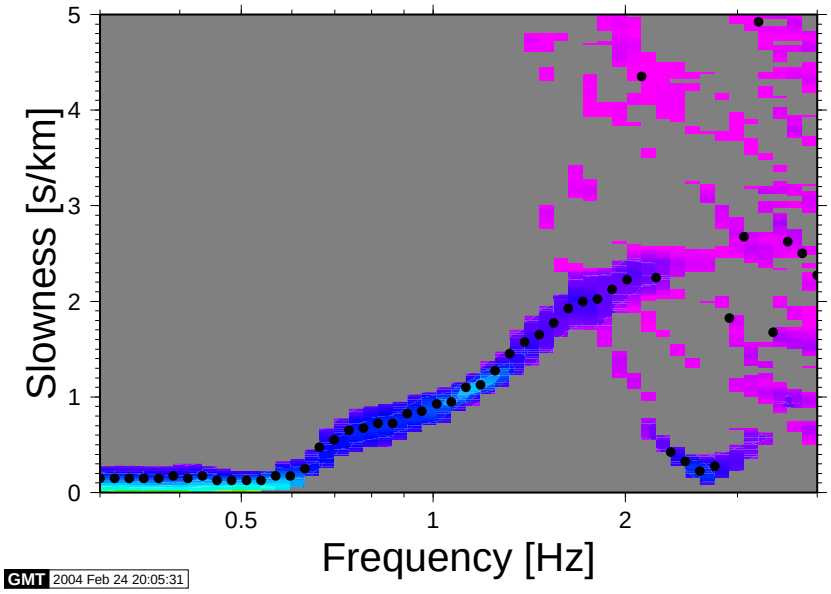

Figure 13: Results of CAPON processing of an array data set acquired in the Lower Rhine Embayment (site Pulheim).

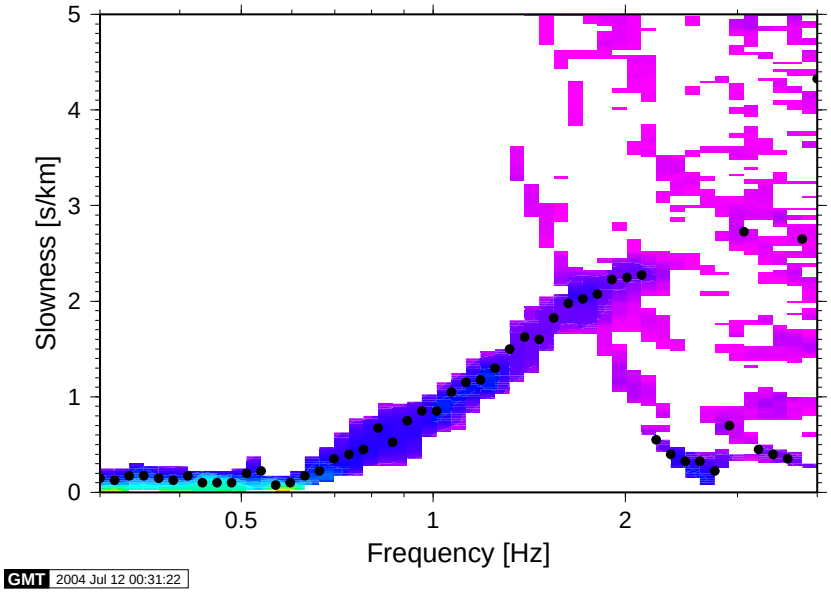

Figure 14: Results of MUSIC2 processing of an array data set acquired in the Lower Rhine Embayment (site Pulheim).

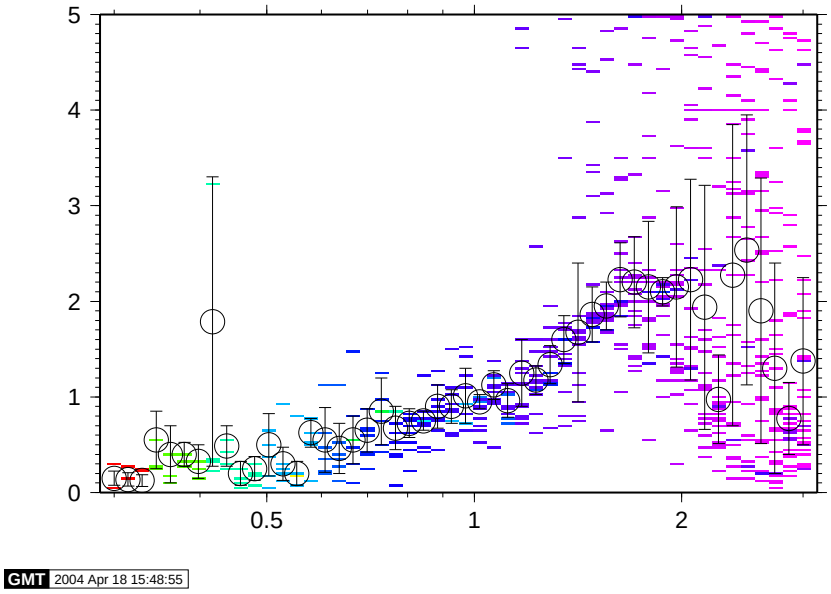

Figure 15: Results of MUSIC processing of an array data set acquired in the Lower Rhine Embayment (site Pulheim).

rings have been selected manually usnig the GEOPSY software package. The ring definition file has been given at the command line to process the data. As a result we obtain 6 autocorrelation curves, one for each ring, as displayed in figure 16.

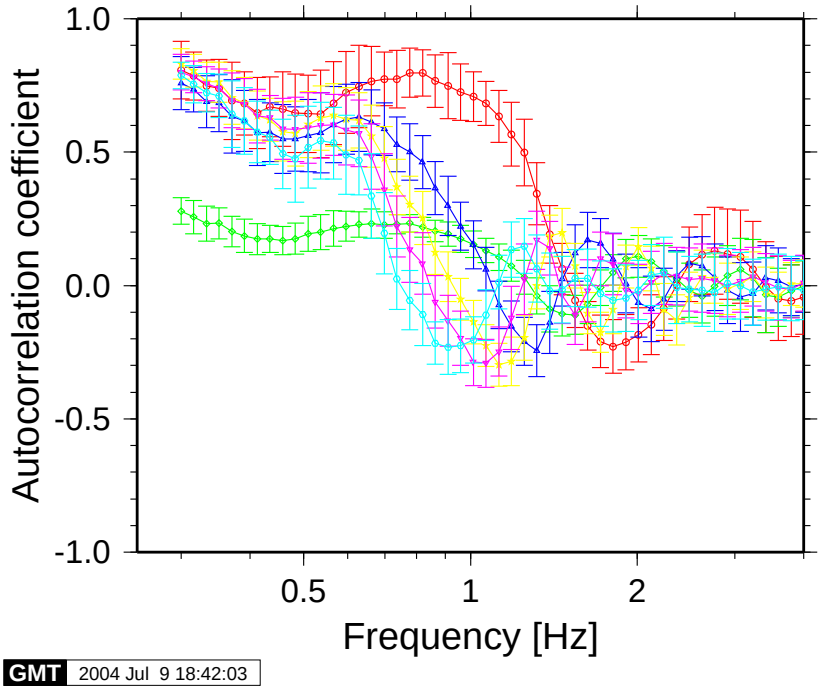

Figure 16: Results of MSPAC processing of an array data set acquired in the Lower Rhine Embayment (site Pulheim). 6 rings have been used here. It is clearly recognized the oszillating nature of the autocorrelation curves. The "oscillation frequency" gets larger for increasing radii of the rings. Thus the smallest ring configuration results in the red curve, the largest in the cyan color. One ring shows a degenerated behavior - in this case we could show that it is related to the very small azimuthal coverage of station pairs coincidencing with very focussed wave propagation directions in the wavefield for frequency bands below 1.2 Hz (see Ohrnberger et al., 2004)

All of the above analysis examples can be reproduced also with the standalone version of **CAP** , called "cap_sa", and the GEOPSY based version "cap". Therefore we will not repeat these examples in the following sections, but just point out the convenient use of "cap" (GEOPSY version) with a GUI-interface and the simple usage of "cap_sa", which does not require to build a database beforehand.

7.3.2 Command line usage with FAKE_DB

"cap_sa" does not require an existing database of any type (neither GIANT nor GEOPSY). It is called from the command line in a similar syntax to cap_giant. Here is the help message, displayed, when cap_sa is called without any command line switch or using the "-h" option.

```
============================================================
USAGE: cap -optflag optarg -optflag optarg ...

optionflags:
        -f: <arg: Last time string, 1997[07[20[01[02[21[.32]]]]]]>
        -l: <arg: Last time string, 1997[07[20[01[03[21[.32]]]]]]>
        -i: <arg: parameter settings file name / see below>
        -g: <arg: station list, separated by '+', e.g. B01+B02+B03+B04>
        -s: <arg: ASCII file containing station coordinates
        -w: <arg: ASCII file containing waveform headers (GSE)>
       [-o: <arg: basename for output files>]
       [-t: <arg: file containing time-frequency boxes for processing>]
       [-a: <arg: file containing MSPAC AC-results> -- starts inversion]
       [-r: <arg: file containing ring definitions for MSPAC processing]
        -h: <noarg: print this Help message>


 Environment variable CAPHOME not set -
 set CAPHOME to home of your cap installation
 and run again - then you'll get a printout of
 a cap sample configuration file (sample.cfg)!

============================================================
```

Two additional, required, option parameters are used for "cap_sa". The "-w" and "-s" switches require a filename as argument. The files must contain the waveform and station information as described in section 7.1. All other options are to be used in analogy to cap_giant. It should be noted, that "cap_sa" is completely platform independent. It can be run on Linux/Solaris systems as well as on Windows or Mac OS platforms.

7.3.3 GUI-interface with GEOPSY DB

This version of **CAP** is called (onyl) "cap" and is the most convenient way for using this software package. Thanks to Marc Wathelet this version contains an easy GUI interface which allows to open an existing GEOPSY database and access the information by just a few mouse-clicks. It is noteworthy that this version provides not only the greatest flexibility with respect to the allowed waveform formats, but it is also fully platform independent.

Calling "cap" on the command line without any argument starts an interactive **CAP** session. In the startup screen (figure 17) a file dialog box is openend in order to allow the user to choose an existing GEOPSY database.

After opening the database, "cap" scans the specified database and extracts the necessary available information for the users and displays a list of station-channel groups defined for this database (figure 18). For the creation of groups within a GEOPSY database, we refer the reader here to the GEOPSY help index. One group must be selected by the user for processing. The plus signs in the tree-like display allow to expand the group and display the information on the individual stations/channels which are contained in the selected group (figure 19).

By highlighting any of the stations, the start and end times of available waveform data will be written to the corresponding fields of the current display. Thus, it is easy to obtain an overview of the availability of waveform data. The user can then edit the start end end times to his/her needs. iIn almost all situations,

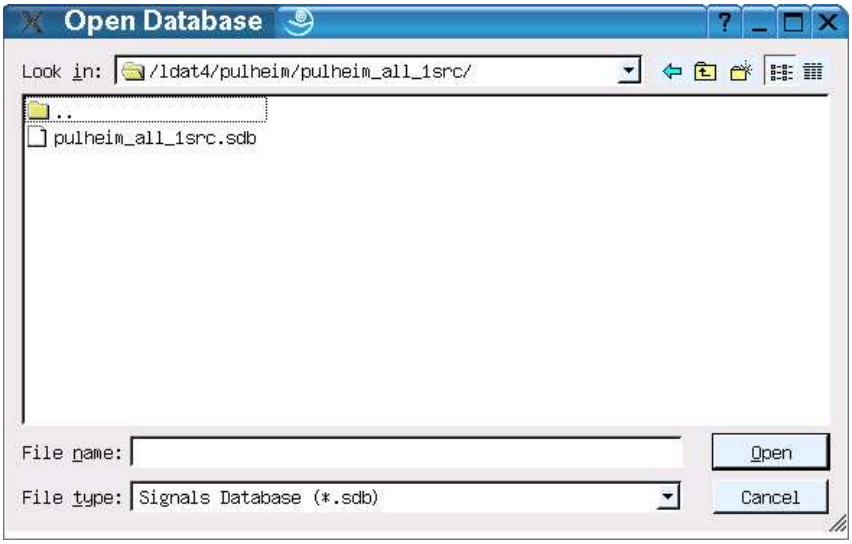

Figure 17: After startup of cap with GUI interface ...

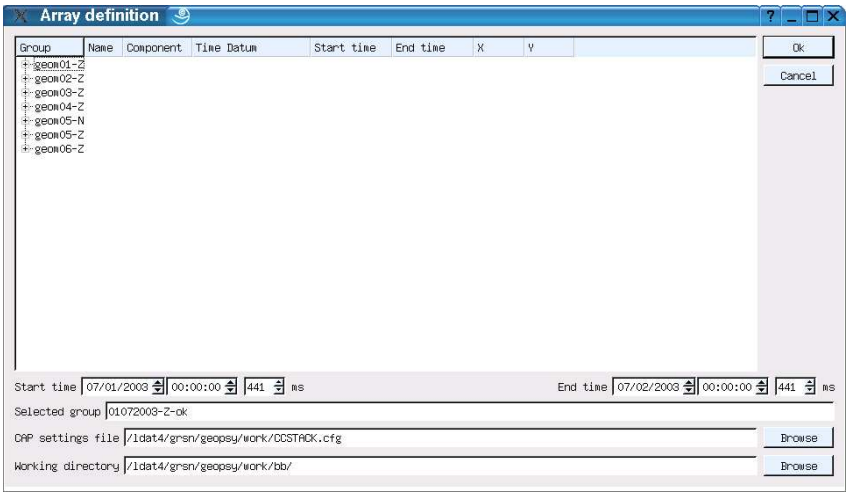

Figure 18: Selecting existing groups for processing.

he/she will try to use the longest time window, in which all stations contain waveform data, for the analysis.

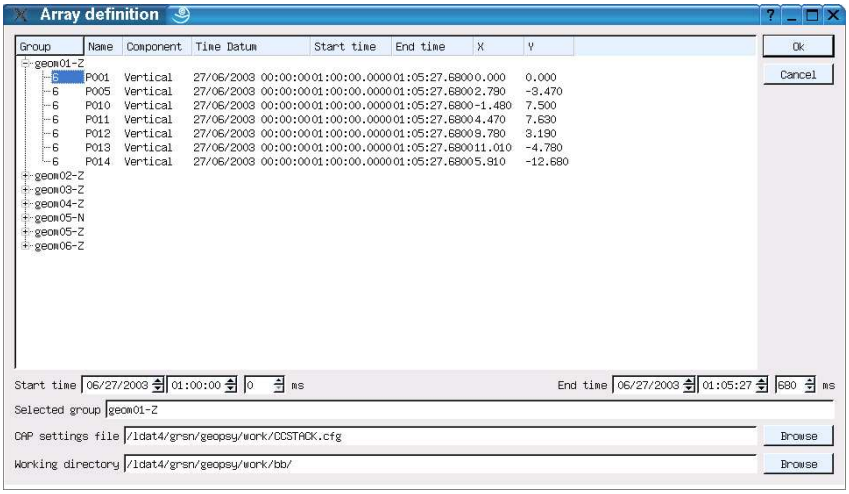

Figure 19: Specifying start and end times.

Once the start and end times have been specified by the user, a configuration file can be chosen in another file dialog box (figure 20).

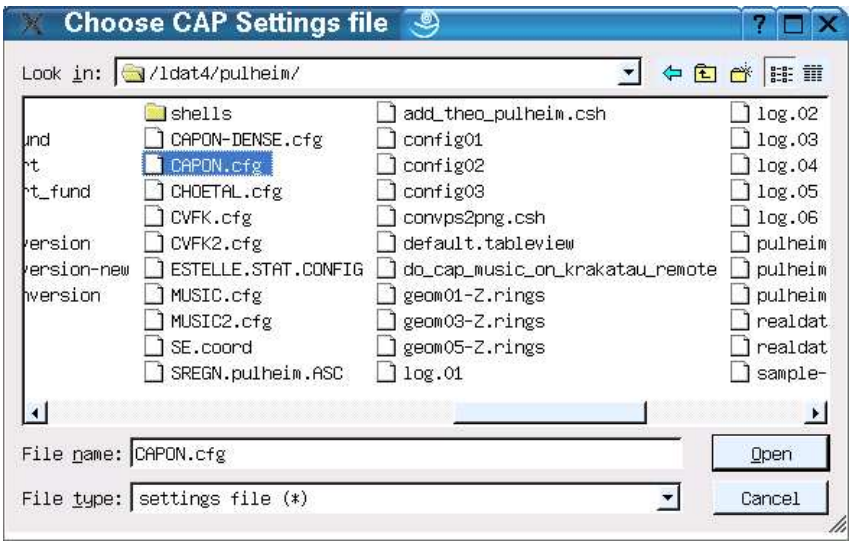

Figure 20: Selecting a configuration file ...

Important: until now, the configuration files have to be edited outside "cap". Finally, a directory has to be chosen by the user where the output files should be written to (figure 21). Please note, that by using the interactive version of "cap" will read the basename of the output files from the setting of variable

OFILE_NAME in the configuration file!

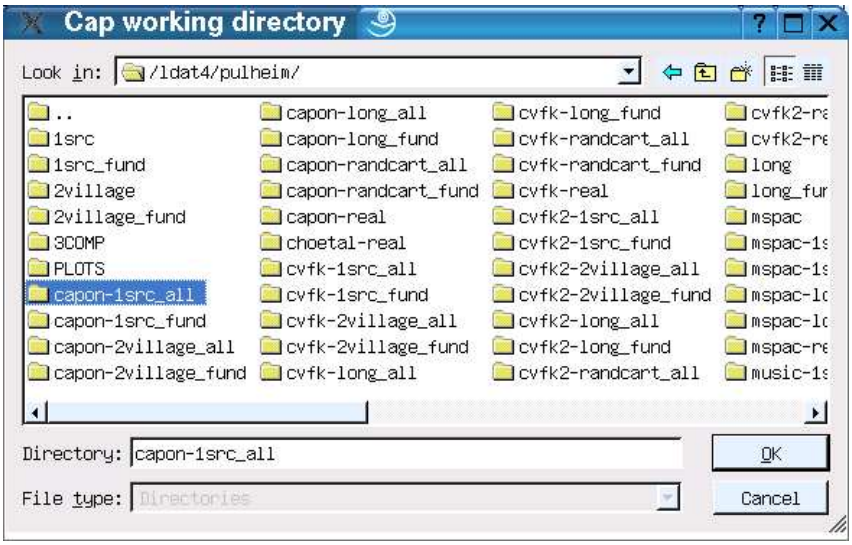

Figure 21: Selecting directory for output files ...

Pushing finally the "OK" button will start the processing according to the settings specified in the chosen configuration file. Easy, isn't it?

7.3.4 Command line interface with GEOPSY DB

If you have used the GEOPSY version of **CAP** , you will have noted, that after start of processing an additional file name "start_cap" has been created. "start_cap" is a small sample script, which allows to reprocess the data in exactly the same way. This is achieved by using "cap" with command line options, similar to cap_giant and cap_sa versions of this software package.

Calling cap with the "-h" option will result in the display of the following help message:

```
Usage:
cap -h -d sdbFile -i parameterFile -g groupName -f startTime -l endTime [-t tfboxFile]
[-r ringFile] [-a mspacFile] [-o outFileBasename]
Without argument options will be asked interactively
        -d database file (relative or absolute path)
        -i parameter file
        -g name of the group containing the array to process
        -f First time string, 1997[07[20[01[02[21[.32]]]]]]
        -l Last time string, 1997[07[20[01[03[21[.32]]]]]]
        -o basename of output files
        -t tfbox file
        -a mspac output file for starting inversion
        -r ring file for mspac computation with visually determined radii
        -h      This help message
```

You will note, that one additional required option appears for the GEOPSY version of **CAP** . Instead of specifiying the database settings via environment variables, as in the cap_giant case, or using the "-s" and "-w" switches which apply only to the cap_sa version, the command line switch "-d" is used to specify the GEOPSY database file. Furthermore, the "-g" option now does not contain a list of station names separated by "+" signs, but simply the group-name of an exisiting station/channel group of the selected database is given. All other command line options behave in exactely the same way as before.

Please note, that the use of the command line interface allows to process data without any user interaction. Using this capability allows for example to rerun the same analysis method on several smaller data windows, or to apply all analysis method to the same data window, or to process the same data with different station geometries, or ..., or ... When using the command line scripting of "cap", please keep in mind, that each run creates (and potentially overwrites) the script file named start_cap. Therefore, it is wise to use another name for your personal script. Enjoy!

8 Future developments

The current state of the software package **CAP** has still to be considered as experimental. However, the main processing routines as there are the f-k analysis methods have reached a relatively stable state now. The single component MSPAC processing is considered to be stable whereas the horizontal component processing and the inversion procedure will probably revised and further improved. The methods described as experimental or supplemental are currently subject to further investigations and may be discontinued or stabilized depending on their potentiality to become useful for ambient vibration analysis.

In general we see the following points as most important for working on within the next months.

- In order to enable a full release of **CAP** under a GPL or GPL like license, it is necessary to replace parts of the code which are published under more restrictive licensing conditions, e.g. FFT computations are mostly performed by routines of Numerical Recipes in C (Press et al., 1992). In future we will migrate to the GPL licensed fftw-3.0.1 package. Similarily the GIANT version of **CAP** makes use of the commercial RAIMA database manager. There, we tend to implement the database structures of GIANT using the GPL licensed mySQL database (http://www.mysql.org).
- Improve certain processing outputs and output file structures. Enable binary file storage and supply utilities to read binary output files.
- Implementation of additional hyopthesis testing strategies for pre-selection of time windows containing dominant surface wave contributions in the wavefield.
- Implementation of plotting script functionality for MSPAC and H/V processing outputs
- Three component modified SPAC implementation and combined inversion for Rayleigh and Love wave dispersion curves and paritioning of seismic energy between both surface wave components.
- Simplification of GIANT/GEOPSY discrepancies, e.g. regarding the internal representations of coordinates in geographical or cartesian coordinates.
- Optimization of computational speed by unwrapping double, triple, ... array memory allocation and addressing through one-dimensional array types with the whole software package.
- Continue bug hunting!

9 About ...

9.1 Copyright

We intend to bring **CAP** under the GNU General Public License. However, as stated above, until now, there are still parts of the code which conflict with this intention.

Especially the GIANT version of this software package can not be made publically available in source code form in the moment, as the underlying database structure and database access functions are based on lbraries of the RAIMA database manager, a product for developing database applications for which the the University of Potsdam has purchased a developer license. We intend to change the database structure to mySQL or other publically available database engines published under the GPL to avoid these restrictions in near future.

For the moment we want to state that we provide this software 'as is" without warranty of any kind. The entire risk as to the quality and performance of the program is with you (the user of this software package). Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.

9.2 Funding

The software package **CAP** has been developed within the context of the SESAME project (EU-Grant No. EVG1-2000-00026) from 05/2001 to 10/2003.

9.3 Acknowledgments

The manual has been written in LaTeX. Both postscript versions and pdf-versions of this document are available on request to mao@geo.uni-potsdam.de

Figures have been produced by GMT, xfig, the picture environment of LaTeX, xgrab, etc. and figures have been partly postprocessed with the Gimp.

I am indebted to the OpenSource community and GNU/GPL related activities.

10 References

References

[Ata04] Atakan, K., Duval A.-M., Theodulidis, N., Guillier B., Bard P.-Y. aand the SESAME-Team. The H/V spectral ratio technique: Experimental conditions, data processing and empirical reliability assessment. *Paper No. 2268, 13th World Conference on Earthquake Engineering Vancouver, B.C., Canada August 1-6*, 2004.

[Aki57] Aki, K. Space and time spectra of stationary stochastic waves with special reference to microtremors. *Bull. Earthq. Res. Inst.*, 35:415-456, 1957.

[Bet03] Bettig, B., P. Bard, F. Scherbaum, J. Riepl, C. Cornou and D. Hatzfeld. Analysis of dense array noise measurements using the modified spatial auto-correlation method (spac). application to the grenoble area. *Bolletino di Geofisica Teorica ed Applicata*, 42(3/4):281-304, 2003.

[Bar98] Bard, P. Microtremor measurements: a tool for site effect estimation?. *Second International Symposium on the Effects of Surface Geology on seismic motion, Yokohama, December 1-3, 1998, Irikura, Kudo, Okada & Sasatani, (eds), Balkema 1999, 3*:1251-1279, 1998.

[Cam03] Campillo, M. and Paul, A. Long range correlations in the diffuse seismic coda. *Science*, 299:547-549, 2003.

[Cap67] Capon, J. and Greenfield, R. J. a. K. Multidimensional Maximum-Likelihood Processing of a Large Aperture Seismic Array. *Proceedings of the IEEE*, 55(2):192-211, 1967.

[Cap69] Capon, J. High-resolution frequency-wavenumber spectrum analysis. *Proc. IEEE*, 57:1408-1418, 1969.

[Fer91] Ferrazzini, V., K. Aki and Chouet B.A. Characteristics of seismic waves composing hawaiian volcanic tremor and gas-piston events observed by a near-source array. *J. Geophys. Res.*, 96:6199-6209, 1991.

[Jur88] Jurkevics, A. Polarization analysis of three-component array data. *Bull. Seism. Soc. Am.*, 78(5):1725-1743, 1988.

[Kv86] Kværna, T. and Ringdahl. Stability of various f-k estimation techniques. *In: Semiannual Technical Summary, 1 October 1985 - 31 March 1986, NORSAR Scientific Report, 1-86/87, Kjeller, Norway*:29-40, 1986.

[Kon98] Konno, K. and Ohmachi, T. Ground-Motion Characteristics Estimated from Spectral Ratio between Horizontal and Vertical Components of Microtremor. *Bulletin of Seismological Society of America*, 88(1):228-241, 1998.

[Lou01] Louie, N. Faster, Better: shear-Wave Velocity to 100 Meters Depth From Refraction Microtremor Arrays. *Bull. Seism. Soc. Am.*, 91(2):347-364, 2001.

[Nei71] Neidell, N. S., a. T. Semblance and other coherency measures for multichannel data. *Geophysics*, 36(3):482-497, 1971.

[Ohr04] Ohrnberger, M., Schissele E., Cornou C., Wathelet M., Savvaidis A., Scherbaum F., Jongmans D. and Kind F. Microtremor array measurements for site effect investigations: comparison of analysis methods for field data crosschecked by simulated wavefields. *Paper No. 0940, XIII World conference on Earthquake Engineering, Vancouver, B.C., Canada, August 1-6*, 2004.

[Ohr01] Ohrnberger, M., F. Scherbaum, K.-G. Hinzen, S.-K. Reamer and B. Weber. Vibrations on the Roll - MANA, a Roll Along Array Experiment to map Local Site Effects Across a Fault System. *Eos. Trans. AGU, Abstract S21D-0606, Fall- Meet. Suppl.*, 82(47), 2001.

[Pre92] Press, W. H., S. A. Teukolsky, W. T. Vetterling and B. P. Flannery. *Numerical recipe in c: the art of scientific computing*, 1992. Cambridge University Press, 1992.

[Rie98] Rietbrock, A. and Scherbaum, F. The GIANT analysis system. *Seismological Research Letters*, 69(6):40-45, 1998.

[Sch03] Scherbaum, F. a., K.-G. a. Hinzen and M. Ohrnberger. Determination of shallow shear wave velocity profiles in the cologne germany area using ambient vibration.. *Geophys. J. Int.*, 152:597-612, 2003.

[Sca98] Scales, J. a. and R. Snieder. What is noise?. *Geophysics*, 63(4):1122-1124, 1998.

[Sch81] Schmidt, R. O. *A signal subspace approach to multiple emitter location and spectral estimation.* Stanford University, Stanford, California, 1981.

[Sch86a] Schmidt, R. Multiple emitter location and signal parameter estimation. *IEEE Trans. on Antennas and Propagation*, 34:276-280, 1986.

[Sch86b] Schmidt, R. Multiple source df signal processing: an experimental system. *IEEE Trans. Ant. Prop.*, 34(3):281-290, 1986.

[Sei80] Seidl, D. The Simulation Problem for Broad-Band Seismograms. *Journal of Geophysics*, 48:84-93, 1980.

[Sha04] Shapiro, N. M. and Campillo, M. Emergence of broadband Rayleigh waves from correlations of the ambient seismic noise. *Geophys. Res. Lett.*, 31(7), 2004.

[Tar82] Tarantola, A., and B. Vallette Generalized nonlinear inverse problems solved using the least squares criterion. *Rev. Geophys. Space Phys.*, 20:219-232, 1982.

[Wax85] Wax, M. and T. Kailath. Detection of signals by information theoretic criteria. *IEEE Transactions on ASSP*, 33(2):387-392, 1985.

[Zyw99] Zywicki, D. J. *Advanced signal processing methods applied to engineering analysis of seismic surface waves.* Georgia Institute of Technology, 1999.

A Sample configuration file

here is some sample configuration file for **CAP**

```
********** processing settings ************************

METHOD  0       # select method of processing:
                0: CVFK         # Sembalnce based conventional FK after
                                # Kvaerna and Ringdahl, 1986
                                # Sliding window analysis - SLOW!
                1: CVFK2        # Conventional FK Beampower analysis
                                # From average complex cross spectral matrix
                                # Not Semblance but power based!
                2: CAPON        # Capon's high-resolution FK
                                # From average complex cross spectral matrix
                                # Power based estimate!
                3: SLANTSTACK   # SLANTSTACK analysis steered on single azimuth
                                # Stacked average from shifted FFT's
                                # Power based estimate
                                  <--- was deleted once, but will be
                                       reconsidered asap
                4: MSPAC        # Modified SPAC
                                # Sliding window analysis
                5: MUSIC        # MUltiple SIgnal Classification
                                # Sliding window analysis
                6: MUSIC2       # MUltiple SIgnal Classification
                                # From average complex cross spectral matrix
                7: HTOV         # computes H/V ratios for given stations
                                # and array/network-wide average
                                  <--- not yet implemented

                # methods for pre-selection of ''useful'' time windows
                # output used as input for methods 0, 3(?), 4, 6(?)
                8: CHECK_EIGSPEC# pre-selection method
                9: HYPTEST      # hypothesis testing for pre-selection
                                # allows combinations of hypothesis testing
                                # routines as specified by HYPMETH

                # experimental methods, not fully tested/explored
                10: CCSTACK     # simple CC-stacks between stations
                                # can be used for ZZ stacks only or
                                # for all combinations (COMP keyword)
                11: QEST        # window based estimation of attenuation
                                # according to Ph.D. Thesis by Daren Zywicki
                12: CHOETAL     # paper from Cho et al. 2003/4 - implemented
                                # as 3-station method for all combinations
                                # available in array configuration

                # 'late' methods
                14: CVFK_FAST   # 'fast' CVFK based on gridless maximization of
                                # slowness map via combined simplex and
                                # simmulated annealing approach
                                # (Press et al., Numerical Recipes)
```

```
********** submethods for HYPTEST **********************

HYPMETH         0       # selects method(s) for hypothesis testing
                        # Requires METHOD set to 9!
                        # Current implementation should be used with
                        # single sub-method testing - specified by integer
                        0: t-f pol.-analysis (array-wide, Jurkevics, 1988)
                        1: pol.-model test Christofferson et al., 1988
                        2: pol.-analysis Vidale, 1986
                        3: t-f 3-C complex trace analysis (Rene et al., 1986)
                        4: t-f energy criteria (ridge+energy, Schissele, 2002)
                        5: t-f smoothed phase stack (Schimmel ++ )
                        6: t-f cross analytic signal coherence measure
                        # so far, only option 0 and 4 are implemented
                        # For the future it is planned to combine different
                        # sub-methods for a joint hypothesis test, then:
                        # argument: list of integer separated by '+' signs
                        # e.g. 0+3+4

********** threshold list for submethods of HYPTEST ****

TFPOLJURK_TH1   0.9     <- linearity threshold for single station, all
                           signals more linear than this value are
                           NOT considered!
TFPOLJURK_TH2   0.      <- percentage of stations required to pass the
                           test above!

PAMLTEST_TH1    xx      <- not yet implemented
PAMLTEST_TH2    xx      <- not yet implemented

PAVIDALE_TH1    xx      <- not yet implemented
PAVIDALE_TH2    xx      <- not yet implemented

TFCOMPLEX_TH1   xx      <- not yet implemented
TFCOMPLEX_TH2   xx      <- not yet implemented

TFENERGY_TH1    0.01    <- relative energy threshold per freq. band
TFENERGY_TH2    0.8     <- percentage of array stations contributing

TFSCHIMMEL_TH1  xx      <- not yet implemented
TFSCHIMMEL_TH2  xx      <- not yet implemented

TFXANSIG_TH1    xx      <- not yet implemented
TFXANSIG_TH2    xx      <- not yet implemented

********** applies just for CVFK and CCSTACK method ***************

PREWHITEN       0       # toggle prewhitening on or off
                        0: toggles off
                        1: toggles on

********** applies just for CVFK+CVFK_FAST method in the moment **************
```

```
DETECT_DOMINANT 0       # toggles detection of single dominant signal
                        # in current window by determination of
                        # eigenspectra characteristics - needs SINGVAL_RATIO

SINGVAL_RATIO   10.     # ratio of first to second eigenvalue
                        # from eigenvalue decomposition of covariance matrix

SLOWRESP        0       # computes slowness response for ideal harmonic waves
                        # centered on previously determined fk-maximum
                        # May be used for postprocessing, but slows down
                        # processing speed

****** applies to CVFK(2), CVFK_FAST, CAPON, MUSIC(2) and MSPAC ********

NUM_BANDS       50      # number of bands for FK or MSPAC

LOWEST_CFREQ    0.3     # center frequency of lowest band

HIGHEST_CFREQ   4.0     # center frequency of lowest band

BANDWIDTH       0.1     # half bandwidth of CVFK or MSPAC bands as fraction of
                        # center frequency - filter (1-bw)*fc <-> (1+bw)*fc

BANDSTEP        -1.     # factor used to multiply center frequency in order to
                        # get to next higher center frequency
                          <-- if set to negative values, BANDSTEP is determined
                              from HIGHEST/LOWEST_CFREQ and NUM_BANDS!

********* applies to CAPON, CVFK2 and MUSIC(2) ********

SPATIAL_SMOOTH  0       # toggle spatial smoothing
                        0: toggles off
                        1: toggles on

******** applies only for MUSIC(2) methods ****************

NSRC_SELECT     0       # selection of number of sources
                        * negative integer: use full solution from
                          nsrc = 1 .... M-1 -> creates LARGE output!
                        * 0: automatic determination with AIC
                        * positive integer .lt. M-1: fixed number of sources

*** applies for MSPAC inversion scheme - may be unnecessary in future ********

OMEGA           -1.     # smoothing for a priori gauss distribution
                        # of model parameters for MSPAC dispersion curve
                        # inversion - if set less than 0 - a priori
                        # information is set to unity matrix

APRIORI         1.      # standard deviation of a priori distribution
                        # of model parameters for MSPAC dispersion curve
                        # inversion - if OMEGA is set less than 0
                        # this parameter is not used...
```

```
CR@1HZ          0.6     # cR(2*PI*f) at f = 1 Hz, Rayleigh wave velocity at 1 Hz
                        # for initial dispersion curve model (MSPAC)

CREXP           0.1     # exponent for initial dispersion curve model
                        # c(2*PI*f) = c(1)*(2*PI*f)^-CREXP

BESSMINARG      0.4     # use this to determine minimum argument of bessel
                        # function for mspac inversion scheme -

BESSMAXARG      3.2     # use this to determine maximum argument of bessel
                        # function for mspac inversion scheme

***** applies to all grid dependent computations *****************
      CVFK(2), CAPON, MUSIC(2), SLANTSTACK

GRID_LAYOUT     0       # select grid layout
                        0: POLAR
                        1: CARTESIAN
                        2: LINEAR
                           <--- provided by M. Wathelet
                                for similar functionality as SLANTSTACK
                                here semblance-based, SLANTSTACK power-based

GRID_TYPE       0       # select grid type
                        0: equidistant sampling in SLOWNESS
                        1: equidistant sampling in APPARENT VELOCITY
                           <--- option 1 NOT recommended

GRID_RESOL      201     # number of grid points in sampling direction
                        # for cartesian grid used for x, y coordinate axis
                        # for polar grid layout used for radial axis

GRID_MAX       5.0      # maximum of grid either app. vel. or slowness
                        # for cartesian grid [-GRID_MAX,GRID_MAX]
                        # for polar grid [0,GRID_MAX]
                          <--- note: polar and linear grids are sampled finer
                           for same GRID_RESOL compared to cartesian grids
                        # slowness/app.vel. resolution polar:
                        #         GRID_MAX/(GRID_RESOL-1)
                        # slowness/app.vel. resolution cartesian:
                        #         2*GRID_MAX/(GRID_RESOL-1)
                        # slowness/app.vel. resolution linear:
                        #         GRID_MAX/(GRID_RESOL-1)

NPHI            72      # number of azimuthal steps for polar grid layout
                          <--- Azimuth resolution = 360/NPHI

LINEAR_PHI      220.    # Backazimuth for steering in case of LINEAR GRID_LAYOUT
                        # value is given in DEGREES as backazimuth -
                        # usual convention (N == 0., E == 90.)

MAPFRAC         0.01    # percentage of highest fk-map values dumped to output

********* applies to CCSTACK method ***************************
```

```
NSTACK          5000

SEED            0       # 0 will select some seed from system clock,
                        # any other value will be used as fixed seed
                        # to start the random number generator
                        # (enables to restart with the same random series)

********* applies to all methods ******************************

COMP            1       # select component to process
                        1: vertical component Z
                        2: north component N
                        3: east component E
                        22: radial component R
                        33: transverse component T
                        123: all three components

WINFAC          10.0    # window length is adjusted to center frequency
                        # of processed frequency band FCENT -
                        # window length is set to:
                        # WINLEN = WINFAC * 1./FCENT
                        # WINFAC set to positive value OVERRIDES
                        # settings for WINLEN and STEP!
                        # Turned off if WINFAC < 0

OVERLAP         -1      # selects amount of overlap depending
                        # on center frequency:
                        # 0 -> STEP = 0.5*WINLEN(HIGHEST_CFREQ)
                        #   ---> may cause highly oversampled
                        #        processing for lower freqs.
                        #   ---> causes long processing times
                        # 1 -> STEP = 0.5*WINLEN(LOWEST_CFREQ)
                        #   ---> may cause gaps in data processing
                        #        for higher freqs.
                        # 0 < OVERLAP < 1
                        #   -> STEP approx.
                        #   0.5*WINLEN(OVERLAP*(HIGHEST_CFREQ-LOWEST_FREQ))
                        #   ---> some compromise in OVERLAP
                        # OVERLAP < 0 -> uses STEP = 0.5*WINLEN(FCENT)
                        #   ---> 50% overlap in all freq. bands

WINLEN          5.0     # window length in seconds
                        # fixed window length for all frequency bands
                        # only if WINFAC is set to negative values

STEP            1.0     # forward step in seconds
                        # only used if fixed window length is selected
                        # (WINFAC set to negative values)

TAPER_FRAC      1.      # fraction for cosine taper
                        # used for all FFT/DFT computations

********** applies for SLANTSTACK and HTOV ******
```

```
POWSPEC         0       # flag whether power spectrum is calculated by
                        # stacking windows or smoothing in spectral domain
                        0: window stacking
                        1: smoothing in Fourier domain

********** applies just for HTOV ******

KOSMOOTH        30      # smoothing parameter b for smoothing window after
                        # Konno & Ohmachi 1998

********** applies just for SLANTSTACK ******

STRIKE          -1.     # strike of line for slantstack analysis
                        # values < 0 indicates use of regression result
                        # from linear array configuration
                        # values >= 0 are interpreted as the LINEAR_PHI parameter

******** I/O settings **********************************

OUTPUT_FILE     test.out        # basename of output file -
                                # extensions are added for output files
                                # this value can be overwritten in the
                                # command line with option '-o'

OFILE_TYPE      0       # flag for output file type
                        0: write out ASCII file
                        1: write out BINARY file
                           ---> header is always ascii

WRITE_TRACES    0       # flag if preprocessed traces should be written out
                        # used for finding errors in preprocessing steps
                        0: don't write out preprocessed traces
                        1: write out preprocessed traces

******** preprocessing parameters **********************

DECIMATE        0       # integer decimation factor - .leq. 1 turns off

SEIDL           0       # flag for instrument simulation
                        0: don't simulate common instrument response
                        1: simulate common instrument response
                           <--- requires instrument response files
                                in GSE1.0 PAZ format - this option
                                is just applicable for cap used with
                                GIANT or in the standalone version (FAKE_DB)

FSIM            0.2     # corner frequency of simulated instrument

HSIM            0.7     # fraction of crit. damping of simulated instrument

BBP_FILTER      0       # flag for butterworth bandpass filtering
                        0: don't filter
                        1: filter
```

```
BBP_LOW         0.1     # lower corner frequency for butterworth bandpass

BBP_HIGH        5.0     # upper corner frequency for butterworth bandpass

BBP_ORDER       2       # number of sections for butterworth bandpass
                          <--- remember: 1 section contains
                               1 conjugate complex pole pair

ZERO_PHASE      0       # flag for zero phase filtering
                        0: just forward filtering
                        1: zero phase filter - forward/backward filtering
                           <--- doubles number of sections!

GAUSSNOISE      0.05    # if value .lt. 0 then gaussian noise is added
                        # to all traces GAUSSNOISE specifies the standard
                        # deviation of gaussian noise as a fraction of
                        # the standard deviation computed for each individual
                        # trace - allows to control fixed signal to noise
                        # ratios for stationary signals

******** more specialized parameters **********************

TIME_CORR       0       # flag if time corrections have to be applied
                        0: don't need time correction
                        1: need time correction
                        # Comment: allows only full sample time shifts!

3DCORRECT       0       # flag whether 3D array geometry is evaluated
                        0: option turned off
                        1: best plane fitted to 3D geometry of array
                        # Comment: this option is only reasonable for arrays
                        # set up on steep slopes, however directions are then
                        # calculated with respect to the gradient of the best
                        # fitting plane --->
                        # this is no longer a ZNE coordinate system!
```